Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Análise da Comunidade Brasileira de Sistemas de Informação Utilizando Diferentes Abordagens de Banco de Dados

Natan de Souza Rodrigues

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Orientadora
Prof.ª Dr.ª Célia Ghedini Ralha

Brasília
2015

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Curso de Computação — Licenciatura

Coordenador: Prof. Dr. Wilson Henrique Veneziano

Banca examinadora composta por:

Prof.ª Dr.ª Célia Ghedini Ralha (Orientadora) — CIC/UnB
Prof.ª Dr.ª Maristela Terto de Holanda — CIC/UnB
Prof. Dr. Edison Ishikawa — CIC/UnB

CIP — Catalogação Internacional na Publicação

Rodrigues, Natan de Souza.

Análise da Comunidade Brasileira de Sistemas de Informação Utilizando Diferentes Abordagens de Banco de Dados / Natan de Souza Rodrigues. Brasília : UnB, 2015.
55 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2015.

1. BD, 2. Comunidade Científica de SI, 3. NoSQL, 4. Rede de Colaboração Científica, 5. SBSI

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Análise da Comunidade Brasileira de Sistemas de Informação Utilizando Diferentes Abordagens de Banco de Dados

Natan de Souza Rodrigues

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Prof.ª Dr.ª Célia Ghedini Ralha (Orientadora)
CIC/UnB

Prof.ª Dr.ª Maristela Terto de Holanda
CIC/UnB

Prof. Dr. Edison Ishikawa
CIC/UnB

Prof. Dr. Wilson Henrique Veneziano
Coordenador do Curso de Computação — Licenciatura

Brasília, 3 de Julho de 2015

Dedicatória

Dedico aos meus pais, tão importantes em minha vida.

Agradecimentos

Agradeço a Deus pela força e saúde durante a composição deste trabalho.

Resumo

Neste trabalho foi realizado uma análise das informações relacionadas à comunidade de Sistemas de Informação (SI) no Brasil. Para viabilizar a análise foi necessário a definição, modelagem e desenvolvimento de um banco de dados capaz de armazenar as informações da comunidade de forma centralizada. Como principal informação da comunidade foram utilizadas as diversas edições do Simpósio Brasileiro de Sistemas de Informação (SBSI), realizadas no período de 2005 a 2014, de onde foram extraídos as publicações de artigos nas trilhas técnicas do evento. Algumas questões de pesquisa foram definidas para limitar o escopo do trabalho, com o lema que hoje rege os membros da comunidade de SI no país - uma comunidade madura é aquela que conhece seus membros e suas respectivas pesquisas. No desenvolvimento do trabalho foram utilizadas duas diferentes abordagens de banco de dados, sendo uma a tradicional abordagem relacional e outra orientada a grafos, com a finalidade de viabilizar a análise da rede de colaboração científica dos pesquisadores da área de SI. Após o desenvolvimento dos bancos, foi feito um estudo comparativo verificando a eficiência de manipulação dos bancos, com o objetivo de verificar as vantagens e desvantagens das abordagens utilizadas.

Palavras-chave: BD, Comunidade Científica de SI, NoSQL, Rede de Colaboração Científica,SBSI

Abstract

In this work, it was developed an analysis of the Information Systems (IS) community in Brazil. In order to make this analysis, the definition, modeling and development of a database to store the community information was necessary. The scientific papers published in the several editions of the Brazilian Symposium of Information Systems (from 2005 to 2014) was used as the main community information. Some research questions were defined in this work according to the IS community current lemma - a mature community is one that knows its members and their research. There was used two different database approaches, the traditional relational model and the graph oriented one, in order to allow the analysis of the IS scientific collaboration network. After the development of both databases, a study to check the efficiency of both databases was conducted to verify the advantages and drawbacks of each approach.

Keywords: DB, IS Scientific Community, NoSQL, Scientific Community Network, SBSI

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

O crescimento de redes sociais e métodos adequados de armazenamento de grande volume de informação, nos leva ao estudo de meios para elaboração e manipulação dessas informações de forma rápida e eficiente. Existe no Brasil uma rede social científica formada pelos pesquisadores da área de Sistemas de Informação (SI), os quais publicam artigos científicos para comunicação de suas pesquisas. Para coletar dados desses pesquisadores foram utilizadas as informações dos artigos publicados nas trilhas técnicas do maior evento de SI do país, o qual é promovido pela Sociedade de Computação (SBC) – o Simpósio Brasileiro de Sistemas de Informação (SBSI). As informações dos anais do evento englobam todas as edições com informação digital, incluindo o período de 2005 a 2014.

Sabe-se que o perfil de uma comunidade pode ser traçado com diversas informações, mas optou-se nesta pesquisa por recuperá-las de uma base de dados relacional centralizada e também de uma base de dados não-relacional, uma vez que acredita-se que estas informações serão muito importantes quando disponibilizadas na Web. Ressalta-se que anteriormente a realização deste trabalho, não existia uma base de dados centralizada ou uma rede social digital, que permitisse a recuperação de informações da rede social científica formada pelos pesquisadores da área de SI.

Durante a confecção deste trabalho algumas perguntas importantes foram feitas para direcional o escopo da pesquisa, as quais também podem ser úteis para os integrantes da comunidade Brasileira de SI. A finalidade em responder as questões definidas é aumentar o conhecimento dos pesquisadores sobre a comunidade a que pertencem, de acordo com o atual lema dos membros: uma comunidade madura é aquela que conhece seus membros e suas respectivas pesquisas. As perguntas preliminares que foram definidas para auxiliar a comunidade a se conhecer melhor são:

- Qual o volume de artigos aceitos nas diversas edições do SBSI?
- Quais os pesquisadores que mais publicam no SBSI?
- Com quem estes pesquisadores publicam?
- Quais as instituições com maior representatividade no SBSI?

Respostas a estas questões certamente auxiliam a tomada de decisão futura de planejamento e organização das edições do SBSI, aumentando o conhecimento da comunidade sobre seus membros e instituições de vinculação. Certamente dominando as respostas a

estas questões, é possível delinear outras questões associadas, as quais também seriam úteis, como por exemplo, saber quais as sub-áreas de SI são mais visíveis para o público. Acredita-se que esta questão está relacionada com as pesquisas publicadas pelos membros em seus artigos no SBSI, sendo que estes autores tem um papel importante na comunidade. Uma tomada de decisão associada a esta informação, seria por exemplo a escolha de crescimento da área através de sub-eventos associados ao SBSI, os quais poderiam ser liderados por estes pesquisadores.

1.1 Problema

Para representação oficial da comunidade de SI no país foi criada em 2010 a Comissão Especial de Sistemas de Informação (CE-SI) na SBC [1], a qual reúne os pesquisadores da área de SI, incluindo acadêmicos, profissionais de mercado, estudantes de graduação e pós-graduação, sendo estes interessados em aspectos teóricos e/ou práticos da área de SI. A CE-SI realiza reuniões plenárias anuais durante o evento do SBSI, onde são discutidas questões de interesse da comunidade, definidos os responsáveis pela próxima edição do SBSI e também as ações para o desenvolvimento de pesquisa na área.

Desta forma, a realização do SBSI anual está sob a coordenação da CE-SI, mas um problema atual é a gestão da informação da comunidade de SI, pois esta tem crescido bastante desde a criação oficial da CE-SI na SBC em 2010. Essa gestão é dificultada pela falta de um repositório digital centralizado, que pelo menos inclua as informações das edições do SBSI. Atualmente os dados de algumas edições constam da Biblioteca Digital Brasileira de Computação – BDBComp, apoiada pela SBC e administrada pelo Laboratório de Banco de Dados – DCC da UFMG. No BDBComp estão registradas sete edições do SBSI incluindo o período de 2008 a 2014[1].

Sabe-se que a realização anual de qualquer evento científico inclui muita variação no número de trabalhos científicos apresentados, consequentemente afetando a participação de autores vinculados a estes trabalhos. Esta variação precisa ser analisada para melhor gerenciamento de um evento com grande influência para a comunidade de SI do país – o SBSI, onde a análise dos dados das edições já realizadas pode auxiliar na definição do perfil da comunidade, através de um levantamento das sub-áreas de pesquisa mais utilizadas, os pesquisadores, as instituições de ensino e pesquisa mais atuantes, enfim, características da comunidade que refletem a sua constituição atual.

Conforme exposto, devido a falta de dados não é possível conhecer a comunidade atual de SI, trazendo prejuízo no processo de tomada de decisão da CE-SI. Ressalta-se que a orientadora deste trabalho tem uma expressiva atuação na CE-SI assumindo a coordenação geral nos períodos de 2013/2014 e 2014/2015, além de ter sido vice coordenadora durante a criação em 2010/2011, podendo portanto testemunhar o lema atual dos membros.

1.2 Objetivos

Capturar as informações de publicações no SBSI, modelar e desenvolver dois bancos de dados utilizando-se abordagens distintas, sendo uma relacional e uma orientado a grafos,

[1] Vide http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/PesquisaEvento?evento=SBSI.

que apresentem de forma digital e centralizada a rede social científica dos membros da comunidade Brasileira de SI.

Como objetivos específicos podemos citar:

- Coletar o maior número dos anais do SBSI em suas diversas edições;
- Estudo de diferentes abordagens de BD para analisar as vantagens e desvantagens de cada uma e poder aplicá-las com dados reais do SBSI;
- Fazer um levantamento e analisar as ferramentas disponíveis para as abordagens, com a finalidade de escolher uma a ser utilizada no trabalho;
- Avaliar o desempenho dos bancos de dados implementados, com foco nas pesquisas de informações sobre a comunidade científica de SI;
- Disponibilizar os resultados alcançados com a implementação dos bancos na Web para viabilizar consultas da comunidade.

1.3 Hipótese

Utilizando as informação da trilha principal do SBSI das edições de 2005 a 2014 é possível mapear a rede social científica da comunidade de SI no país? Tal rede social científica vai auxiliar no processo de tomada de decisão da CE-SI e aumentar o conhecimento dos membros que formam a comunidade Brasileira de SI.

1.4 Organização do Documento

No Capítulo 2 será feita uma introdução teórica incluindo conceitos relacionados a este trabalho e justificando sua utilização. No Capítulo 3 será apresentada a metodologia utilizada, desde o seu planejamento até a execução durante todo o trabalho. Em seguida, no Capítulo 4, as experimentações e análises serão apresentadas, a fim de elucidar o alcance dos objetivos propostos no trabalho. No Capítulo 5 serão discutidos os resultados alcançados e sugestões de trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capitulo será apresentado uma breve introdução das áreas que serviram como fundamentos deste trabalho, incluindo as duas abordagens de banco de dados, relacional e orientada a grafos, rede social científica e em específico a rede formada pela comunidade de SI através dos artigos técnicos publicados no SBSI. Também alguns trabalhos correlatos foram brevemente apresentados.

2.1 Modelo Entidade Relacionamento

Quando Peter Chen [5] formulou a proposta do modelo Entidade-Relacionamento (E-R), baseou-se não na visão de um sistema de aplicação como princípio e sim na compreensão da realidade em que se situava o problema. Neste sentido, faz-se necessário primeiramente responder a seguinte questão: Como projetar um sistema se não entendemos o negócio para o qual será utilizado? Chen dedicou-se a destacar a importância de reconhecer os objetos que compõem este negócio, independentemente de preocupar-se com formas de tratamento das informações, procedimentos, programas, entre outros. Os objetos que precisam ser conhecidos e modelados para um sistema, Chen classificou em dois grupos: Entidades e Relacionamentos. Na sequência será discutido cada um destes grupos.

2.1.1 Entidade

De acordo com [18], no modelo E-R define-se Entidade como aquele objeto que existe no mundo real com uma identificação distinta e com um significado próprio. São as "coisas"que existem no negócio, ou ainda, descrevem o negócio em si. Se alguma "coisa"existente no negócio nos proporciona algum interesse em mantermos dados (informações armazenadas sobre algo), isto a caracteriza como uma Entidade do negócio.

Esta Entidade será então um conjunto de dados em nosso modelo conceitual. É importante destacar que uma Entidade é a representação de uma Classe de Dados do negócio, um conjunto de informações de mesma característica, e suas instâncias (ocorrências), são a representação destes dados. Quando fala-se sobre Classe de Dados, na realidade se está trabalhando mentalmente um nível macro de informações, está-se atuando com abstrações interpretadas de acordo com o meio em que se localiza os interesses e os objetivos

organizacionais. Pode-se ilustrar a representação de uma Entidade no modelo E-R através de um retângulo, com o nome desta Entidade em seu interior, como na Figura 2.1.

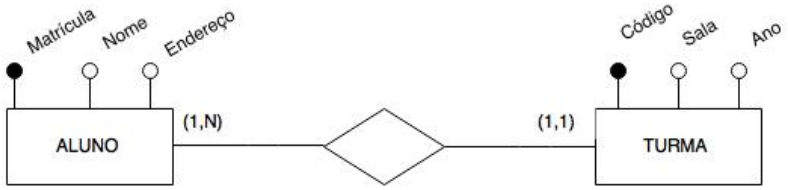

Figura 2.1: Exemplo Modelo Entidade - Relacionamento, onde há as entidades *Alunos* e *Turma*

As instâncias de uma Entidade não são representadas no Diagrama de E-R (DER), mas são semanticamente interpretadas. Para conceituar de forma concreta as Entidades devemos primordialmente nos orientar pelo mundo real, onde acontecem coisas, buscando obtermos domínio do problema, enxergá-las sem a preocupação da construção de um sistema, sem imaginar programas e sim exclusivamente preocupados em retratar uma realidade. No dia-a-dia encontramos e nos deparamos com casos que nos leva a atribuir valores as Entidades. Por exemplo, quando vamos alugar uma casa, procuramos um corretor de imóveis, assim estamos consultando a Entidade "Casa", alugada por um "Corretor", que possui várias instâncias de casa, ou seja, existem várias casas para serem alugadas por aquele corretor.

2.1.2 Atributo

São os Atributos que descrevem as propriedades de uma Entidade. De acordo com Heuser [22], esses Atributos e seus valores juntos descrevem as instâncias de uma Entidade. Considere uma empresa, onde uma Entidade chamada Pessoa que é um objeto sobre o qual deseja-se manter informações armazenadas. O que descreve Pessoa? Pessoa é descrito por um CPF, um nome e sua data de nascimento, como é representado na Tabela 2.1. Poderíamos ainda descrevê-lo com mais dados, tais como nome, endereço ou sexo. Estes dados que caracterizam o objeto pessoa são os Atributos inerentes à Entidade Pessoa.

Tabela 2.1: Exemplo de Atributos que especificam a Entidade Pessoa.

CPF	Nome	Data de nascimento
863945098-18	Manoel Rodrigues Martins	06/06/1944
948530987-98	Antônio Edilson Ferreira	20/07/1969
385739058-14	Davi Pereira Cruz	09/11/1989

Cada instância de Pessoa, cada existência de um objeto da Classe Pessoa, será formada por valores nestes Atributos, sendo que é o conjunto destes valores representados que devemos visualizar em uma tupla. Os valores de um determinado Atributo, nas ocorrências desta Entidade, podem ser diferentes para cada instância, caracterizando a não existência de objetos repetidos dentro de uma Entidade. Estes Atributos, cujos valores nunca se repetem, sempre têm a função de atuarem como identificadores únicos das instâncias

da Entidade. Dentro da abordagem relacional de banco de dados, denominando-se esta propriedade como chave primária de uma tabela.

2.1.3 Relacionamento

Relacionamento é a interação entre os objetos que indicam a dinâmica dos negócios. De acordo com [18], os Relacionamentos são identificados por verbos porque representam as ações que uma Entidade exerce sobre outra. O entendimento sobre o que são efetivamente Relacionamentos e a capacidade de enxergar estes objetos, como participantes do mundo real, são fatores primordiais para que se efetue o trabalho de modelagem de dados com compreensão do que está sendo realizado.

Para um retrato dos objetos e fatos de um problema, os Relacionamentos são os elementos que nos dão o sentido da existência destes objetos e suas inter-relações, sem as quais fica extremamente complexo o entendimento do domínio do problema. No nosso dia-a-dia, tanto em nossas atividades profissionais como nas atividade pessoais, convivemos com os mais variados tipos de Entidades (objetos reais), sendo estes descritos por uma série de Atributos, e que expressam uma realidade de existência. Estas Entidades do dia-a-dia não estão soltas, desligadas umas das outras, e sim relacionadas de forma a mostrar a realidade com um conteúdo lógico, por exemplo:

- As Pessoas *MORAM* em Casas
- As Casas *FORMAM* Ruas
- As Ruas *LOCALIZAM-SE* em Bairros
- Os Bairros *ESTÃO* em uma Cidade.

Cardinalidade

Quando existe um Relacionamento entre duas Entidades, ou seja, o número de ocorrências de uma Entidade que está associado, com ocorrências de outra Entidade, determina a cardinalidade do Relacionamento. Em [22] encontramos a citação que o mundo real apresenta-se com três possibilidade de relacionarmos os dados:

1. Relacionamento Um-para-um (1:1) – cada elemento de uma Entidade relaciona-se com um e somente um elemento de outra Entidade.
2. Relacionamento Um-para-muitos (1:n) – cada elemento da Entidade 1 relaciona-se com muitos elementos da Entidade 2, mas cada elemento da Entidade 2 somente pode estar relacionado a um elemento da Entidade 1.
3. Relacionamento Muitos-para-muitos (n:n) – onde várias entidades A se relacionam com várias entidades B.

2.2 Banco de Dados Relacional

Um bando de dados relacional é um banco que armazena os dados utilizando tabelas bidimensionais. Esse banco veio como uma boa alternativa para a substituição dos banco

de dados hierárquicos que eram utilizados para estruturas simples, mas ao se depararem com estruturas complexas, apresentavam limitações. O banco de dados relacional foi então desenvolvido para suprir essa necessidade e para ser acessado com facilidade pelos usuários, pois a estrutura com tabelas e relacionamentos são de fácil abstração, assim utilizado em variadas abordagens, escopos e aplicações. Após a sua disponibilização, o banco de dados relacional foi fundamental para o sucesso de grandes empresas que apostaram nessa abordagem, como a Oracle e a IBM.

Como base de sua implementação, o banco de dados relacional utiliza o modelo relacional, que foi desenvolvido em 1970 pelo cientista inglês Dr. Edgar Frank Codd (Figura **??**)[6]. Esse modelo de banco de dados é o mais utilizado atualmente, servindo de base para a construção da maioria das aplicações computacionais que utilizam banco de dados.

Juntamente com o modelo relacional, Codd também criou uma série de leis de normalização para os bancos de dados relacionais, buscando que sua ideia de banco de dados relacional fosse comercializada e disponibilizada no mercado em uma forma única, diferente do que vinha ocorrendo. Assim, em 1985, Codd disponibilizou as regras de normalização para serem seguidas na elaboração de um banco de dados [8, 7], conforme segue:

Regra Zero – para ser um sistema de gerenciamento de banco de dados relacional (SGBD), o sistema precisa usar suas facilidades de relacionamento (exclusivamente) para gerenciar o banco de dados. Sendo a regra fundamental uma entre as 12 regras criadas.

Regra Um – as informações no banco de dados devem estar representadas de apenas uma forma, nomeados por valores em posições de colunas dentro de registros de tabelas.

Regra Dois – Todos os dados devem estar acessíveis. Sendo que todo valor na base de dados deve ser logicamente endereçavel por um nome específico do conteúdo tabela, o nome do conteúdo da coluna e o valor da chave primária do conteúdo registro.

Regra Três – o SGBD deve permitir que os campos possam ser nulos ou vazios. Representando uma "falta de informação e informações inaplicáveis"que é sistemática, diferente de todos os valores regulares, e independente de tipo de dados.

Regra Quatro – os metadados devem ser gerenciados e armazenados como dados comuns no banco de dados, ou seja, em tabelas no interior do banco de dados. Também devem estar disponíveis somente aos usuários autorizados, utilizando a linguagem de consulta padrão.

Regra Cinco – o sistema gerenciador pode suportar várias linguagens, sendo que dentre as linguagens, uma deve ser declarativa bem explicitada. Com suporte a definição de dados, definição de visualização, manipulação de dados, segurança e autorização, e gerenciamento de transações.

Regra Seis – as visualizações que são teoricamente atualizáveis deve ser atualizadas pelo sistema.

Regra Sete – o sistema pode fornecer suporte à configuração do nível de operações de *insert*, *update*, e *delete*. As operações de inserção, atualização, e exclusão devem ser apoiadas para qualquer conjunto que pode ser pesquisado e não apenas para uma única linha de uma tabela.

Regra Oito – os aplicativos e recursos *ad hoc* não são afetados logicamente quando os métodos de acesso ou as estruturas de armazenamento físico são alteradas.

Regra Nove – os aplicativos e recursos *ad hoc* não são afetados logicamente quando há alterações de estruturas de tabela que preservam os valores originais da tabela. Alterações nas relações e nas visualizações devem causar pouco ou nenhum impacto nas aplicações.

Regra Dez – todas as restrições de integridade devem ser especificadas separadamente dos programas de aplicação e armazenadas no catálogo. Sendo necessário que seja possível mudar estas restrições sem que necessariamente modifique as aplicações.

Regra Onze – os aplicativos e usuários não devem ser afetados quando o banco de dados altera o armazenamento dos dados, e.g., sistemas distribuídos.

Regra Doze – se o SGBD dá suporte a acesso de baixo nível aos dados, não deve haver um modo de negligenciar as regras de integridade do mesmo.

O modelo relacional, primeiramente, para a sua composição e representação dos dados utiliza tabela com linhas e colunas. Estas tabelas representam relações matemáticas de maneira uniforme, assim facilitando a manipulação dos dados. As relações são compostas por esquemas e instâncias, sendo o esquema um objeto invariável, que define o nome da relação e colunas. A instância representa as tuplas da relação em determinado momento, sendo esta variável de acordo com o domínio definido pelo esquema da relação. Pode-se ver na Figura 2.2 um exemplo de instância de uma relação, sendo que esta deve obedecer um esquema. Na figura a relação é representada por uma tabela.

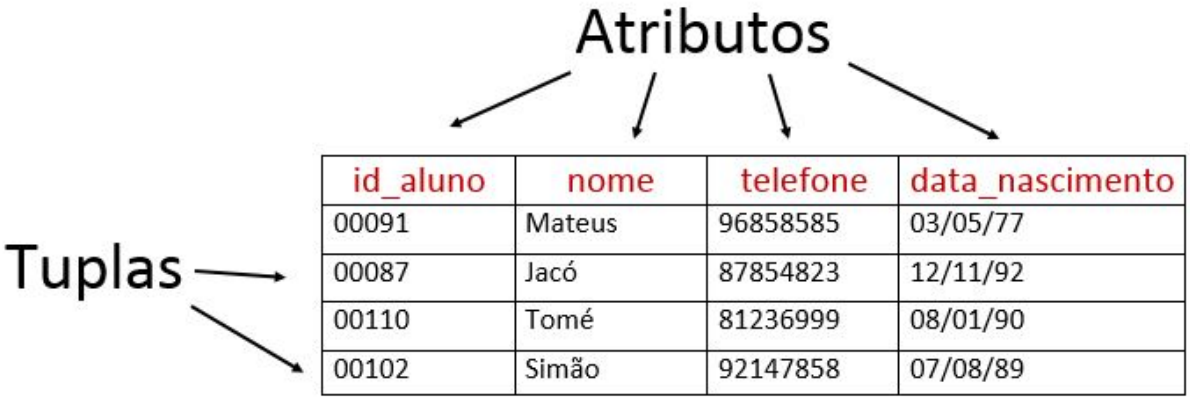

Figura 2.2: Exemplo de uma tabela com atributos e tuplas.

De acordo com [18] a definição clássica de modelo relacional está ligada as operações que são executadas sobre as tabelas, sendo estas operações executadas por linguagens com

fundamentos em álgebra relacional e teoria dos conjuntos. Isso nos permite a manipulação dos dados sem a relevância de identificar a origem e o estado dos dados no banco.

Essa manipulação das tabelas é feita com os dados alocados em linhas e colunas, assim utilizando como ferramenta para manipulação a álgebra relacional, que conta com conjunto de operadores e funções de alto nível. Sendo a Entidade um agregado de dados atribuído a uma caracteristica em comum. A estrutura do modelo relacional:

- Tabelas – armazenam os dados do banco de dados. A tabela é estruturada por linhas e colunas, onde cada linha contém o mesmo conjunto de colunas das demais linhas da tabela. As tabelas se interligam por meio de relacionamentos, associando os atributos de uma tabela com os atributos de outra tabela. Exemplo: A tabela *Pessoa* se relaciona com a tabela *Usuário*, pois todo usuário é uma pessoa.
- Tupla – é cada lista ordenada de colunas da tabela, sendo que alguns atributos dessa linha não precisam ser necessariamente preenchidos, assim assumindo valores nulos.
- Atributo – é a coluna da tabela, que representa certa característica da tabela. Em uma tabela *Pessoa* pode existir colunas como: *Nome* e *Endereço*.
- Chave e Índice
  - Chave – é um conceito lógico que especifica um dado para ser utilizado em determinada busca ou consulta. Sendo subdividido em quatro tipos de chaves.
    1. Chave primária – é um atributo de uma tabela utilizado para identificar uma determinada tupla.
    2. Chave secundária – é utilizada para busca, como uma segunda chave primária. Identifica uma busca, onde deseja-se recuperar dados que tem atributos em comum. Também é utilizada para se ter eficiência no acesso à tabela e à recuperação de um determinado campo.
    3. Chave candidata – é a chave que pode ser utilizada como indentificador único, onde várias colunas ou tuplas podem ter essa característica.
    4. Chave estrangeira – é o atributo principal para ligação entre tabelas, onde é um dos pilares da estrutura do modelo relacional. Quando há um relacionamento entre duas tabelas, sendo estas ligadas por um atributo em comum, em uma das tabelas este atributo é chave primária e na outra tabela é chave estrangeira, caracterizando uma ligação lógica entre as tabelas. Na Figura 2.3 podemos ver que a ligação entre as tabelas se dão pelas chaves primárias(PK) e as chaves estrangeiras(FK), onde a chave primária *OrderID* é uma chave estrangeira na outra tabela interligada.
  - Índice – é um atributo físico que tem como objetivo a otimização da busca da informação na base de dados. Um índice pode estar relacionado com chave, onde uma chave pode ser um índice, mas um índice não é necessariamente uma chave.

De acordo com [18] podemos relacionar as características e vantagens de se utilizar um banco de dados que utiliza uma abordagem relacional:

- Independência total dos dados;

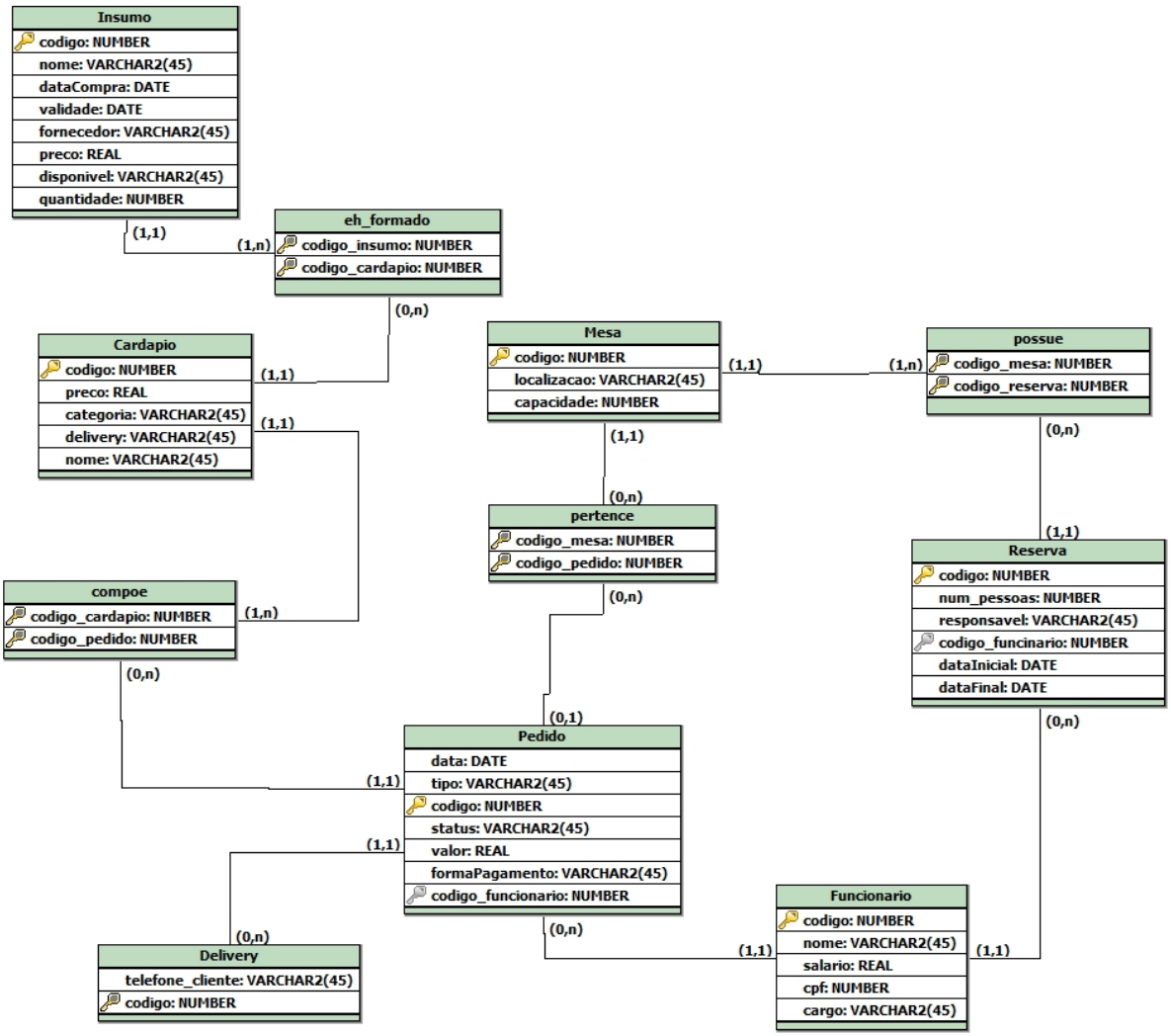

Figura 2.3: Representação de um modelo relacional de banco de dados [19].

- Visão múltipla dos dados;
- Melhor comunicação entre Centro de Processamento de Dados e usuário;
- Redução acentuada na atividade de desenvolvimento de aplicações e o tempo gasto em manutenção.

2.2.1 Linguagem SQL

Para realizar uma consulta no banco de dados deve existir uma linguagem adequada. Os bancos relacionais em sua maioria utilizam a linguagem *Structured Query Language (SQL)* para consulta. De acordo com [30], a linguagem SQL foi criada a partir da álgebra relacional, que é uma linguagem primitiva que auxilia na explicação e fundamentação do modelo de banco de dados relacional. Por ser uma linguagem declarativa, SQL é caracterizada por necessitar que o usuário informe o resultado desejado a ser pesquisado

no banco, diferentemente de uma linguagem procedural, na qual o usuário descreve os passos para chegar ao resultado.

Foi desenvolvido e apresentado pela IBM em 1974, primeiramente com o nome de SEQUEL (*Structured English Query Language*) [4], a qual servia de linguagem para a pesquisa em um protótipo de banco de dados relacional, também apresentado pela IBM no mesmo ano. A IBM lançou no mercado um banco de dados relacional que utilizava uma nova versão da linguagem declarativa lançada três anos antes. Essa nova linguagem passou a ser chamada de SQL, sendo uma compressão do nome SEQUEL da versão anterior. Já em 1979, responsáveis pelo desenvolvimento do sistema lançado pela IBM em 1977, criaram o primeiro SGBD relacional comercial, chamando-o de Oracle.

Esse sistema era o maior concorrente do sistema lançado pela IBM, que liderou o mercado por um bom tempo, pois a marca IBM tinha força em comparação com outras empresas. A linguagem SQL se tornou padrão de mercado e passou a ser utilizada por várias empresas, sendo essas empresas responsáveis pela criação de vários SGBDs. Com esse crescimento da utilização e grande variação de empresas que utilizavam SQL e modelo de dados relacional, viu-se a necessidade de uma padronização da linguagem, pois a quantidade de funções e derivações da linguagem também cresceu, dificultando a utilização do SQL.

Em seguida, o ANSI (*American National Standards Institute*) fez um trabalho de padronização que ficou conhecido como SQL/86. Após esse primeiro padrão, o ANSI juntamente com a ISO (*International Standards Organizations*) criou o padrão SQL/89, que apresentava melhorias em relação a versão anterior. Em 1992, uma nova expansão dessa padronização foi lançada com o nome de SQL/92. Já em 1999 foi lançada uma versão oficial nomeada de SQL/99. No ano de 2003, foi lançada uma atualização, onde a linguagem oferecia suporte a funcionalidades relacionadas a linguagem XML (*eXtensible Markup Language*). Em 2008, uma nova atualização foi lançada, esta oferecia novas funcionalidades como a declaração MERGE e o TRUNCATE TABLE. A sétima revisão do SQL foi lançada em 2011, sendo o suporte a banco de dados temporal a sua principal novidade.

Dividida em subconjuntos, a linguagem SQL conta com palavras e comandos necessários para executar as instruções de manipulação, definição, consulta, administração, controle e verificação do banco de dados:

- DML (*Data Manipulation Language*) – instruções necessárias para manipulações de dados como inserções e exclusões de dados. Algumas instruções são: DELETE, UPDATE, INSERT.
- DDL (*Data Definition Language*) – comandos necessários para definir algumas das estruturas do banco de dados, como tabelas e campos. Alguns comandos são, CREATE e ALTER.
- DCL (*Data Control Language*) – instruções para controle relacionados aos acessos de perfis e usuários ao banco de dados. Exemplos de comandos: REVOKE e GRANT.
- DQL (*Data Query Language*) – este tipo de instrução é necessária para realizar consultas no banco. Conta apenas com o comando SELECT, onde ele é utilizado juntamente com outras cláusulas, necessárias para o filtro dos dados a serem pesquisados. Algumas das cláusulas utilizadas juntamente com o SELECT são: ORDER

BY, FROM, WHERE. Onde essas cláusulas representam algumas condições para a projeção de dados que encontramos comumente na álgebra relacional. O SELECT também trabalha com alguns operadores relacionais como os destacados na Tabela 2.2:

Tabela 2.2: Operadores relacionais.

Operadores lógicos e relacionais	Descrição
<	menor
>	maior
<=	menor ou igual
>=	maior ou igual
=	igual
<>	diferente
BETWEEN	entre
LIKE	procura um padrão
IN	dentro
AND	e
OR	ou

- DAL (*Data Administration Language*) – são comandos utilizados para analisar e verificar a perfomance do banco de dados. São utilizados na administração do banco de dados, desde a instalação até a execução do banco. Alguns exemplos de comandos: START AUDIT, STOP AUDIT.
- DTL (*Data Transation Language*) – abrange as instruções das transações do banco de dados, necessárias para verificar as modificações, importante para garantir a integridade do banco de dados. Das principais instruções destacamos COMMIT e ROLLBACK.

A linguagem SQL alcançou uma grande dimensão, sendo atualmente a mais utilizada na maioria dos banco de dados, seja na área comercial ou na acadêmica. Essa abrangência possibilitou a criação de derivações dessa linguagem por algumas corporações. Mesmo com a padronização definida, a maioria dessas derivações contam com comandos reservados e atribuídos para atender demandas específicas de tais corporações.

2.2.2 MySQL

Um dos sistema de gerenciamento de banco de dados relacional mais utilizados no mundo, o MySQL, utiliza como interface a linguagem SQL. Foi desenvolvido na Suécia em 1995, por David Axmark, Allan Larsson e Michael "Monty"Widenius , que fundaram a MySQL AB, sendo lançado em 1996. Em 2008, a MySQL AB foi comprada pela *Sun Microsystems Inc.* e em 2009 a Oracle adquiriu a *Sun Microsystems*. Assim a Oracle é a nova detentora do MySQL. A popularidade do MySQL se dá pela facilidade de manipulação e funcionalidade para aplicações Web e a sua integração com linguagens, como por exemplo o PHP.

O SGBD MySQL é multi-plataforma, podendo ser instalado em diversos sistemas operacionais. Por utilizar a linguagem SQL como base, o MySQL aceita vários padrões

definidos pela ANSI. Mas ele também conta com comandos próprios que só são encontrados no MySQL. Existem algumas características do MySQL que o torna diferente e preferível por grande parte da comunidade de banco de dados, tais como:

- portabilidade (suporta praticamente qualquer plataforma atual);
- compatibilidade (existem drivers ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como Delphi, Java, C, Visual Basic, Python, Perl, PHP, ASP e Ruby);
- bom desempenho e estabilidade;
- pouco exigente quanto a recursos de novos hardware;
- facilidade no manuseio;
- é Software Livre com base na GPL (entretanto, se o programa que acessar o MySQL não for GPL, uma licença comercial deverá ser adquirida);
- contempla a utilização de vários *Storage Engines* como MyISAM, InnoDB, Falcon, BDB, Archive, Federated, CSV, Solid, entre outros;
- suporta controle transacional;
- suporta *Triggers*;
- suporta *Cursors* (*Non-Scrollable* e *Non-Updatable*);
- suporta *Stored Procedures* e *Functions*;
- replicação facilmente configurável;
- interfaces gráficas (MySQL Toolkit) de fácil utilização cedidos pela MySQL Inc.

Existem vários livros e artigos que especificam e auxiliam no aprendizado do MySQL. Em [30] encontramos o histórico e a conceituação desse sistema. O *download* do MySQL pode ser feito em: `www.mysql.com`. Neste trabalho, [35] foi adotado como referência bibliográfica e auxílio na confecção do banco de dados. Ele oferece uma visão bem ampla do MySQL e ao mesmo tempo é específico em algumas funcionalidades desse SGBD.

2.3 Banco de Dados não Relacional

O conjunto de banco de dados que não utilizam modelo relacional é denominado Not Only SQL (NoSQL). Por diferenciar-se do modelo relacional, que foi muito difundido e utilizado por anos, o NoSQL quebra o paradgima de banco de dados típico, onde se tem um banco de dados diferente, que não utiliza tabelas fixas e não utiliza a linguagem SQL como instrução de pesquisa [34].

O termo NoSQL foi primeiramente utilizado em 1998, como o nome de um banco de dados relacional de código aberto, que não possuía uma interface SQL. Seu autor, Carlo Strozzi [34], alega que o movimento NoSQL *"é completamente distinto do modelo relacional e portanto deveria ser mais apropriadamente chamado "NoREL"ou algo que produzisse o mesmo efeito"*.

Nos últimos anos, o banco de dados NoSQL teve maior procura pelas tendências computacionais, como redes sociais e a computação em nuvem, as quais necessitam de escalabilidade horizontal. Os bancos de dados NoSQL atendem estes requisitos no armazenamento de dados. Isso nos mostra que o NoSQL serve de alternativa para vários ambientes computacionais, assim mostrando que tem efetividade.

Em determinados modelos de banco de dados NoSQL, como o orientado a documentos, toda informação necessária estará agrupada no mesmo registro, ou seja, em vez de se ter o relacionamento entre várias tabelas para formar uma informação, ela estará em sua totalidade no mesmo registro. Grandes empresas como o Facebook e a Google utilizam esse tipo de banco de dados e contribuem para difundir a utilização do NoSQL. Essa utilização nos mostra também que o objetivo do NoSQL não é a eliminação e diminuição do uso do modelo relacional, mas servir como alternativa ao modelo, pois a aplicação de modelo relacional ás vezes não se encaixa em determinadas aplicações, sendo necessária a utilização de NoSQL.

Durante a execução deste trabalho alguns projetos de banco de dados NoSQL foram pesquisados. De acordo com [15], são bancos de dados NoSQL:

- CouchDB;
- MongoDB;
- Db4o;
- Redis;
- SimpleDB;
- Hbase;
- Tabular;
- Cassandra;
- Hypertable;
- Neo4j;
- InfiniteGraph; e
- Mnesia.

2.3.1 Banco de Dados Orientado a Grafos

Sendo um dos modelos contidos nos bancos de dados NoSQL e diferentemente dos bancos de dados relacionais, os bancos de dados orientados a grafos contam com relacionamentos mais naturais. Nesse modelo existem estruturas formadas por arestas (relacionamentos) e vértices (nós), nos quais os dados são armazenados [32]. De acordo com [3], um banco de dados relacional pode ser transformado em um banco de dados modelado à grafo, onde os grafos de dependência para as entidades são transformados em grafos estrela. Esse modelo de grafos estrela é transformado em um modelo de hipergrafo para o banco de dados relacional, que, por sua vez, pode ser usado para desenvolver um modelo de relação de domínio transformando-se em um modelo orientado a grafo.

O armazenamento de dados em um banco de dados NoSQL é estruturado armazenando dados em nós e relacionamentos, e.g., em uma família temos as pessoas e elas são ligadas pelo grau de parentesco, onde o grau de parentesco são os relacionamentos e as pessoas são os nós. Podem existir vários tipos de relacionamentos e vários tipos de nós. Assim, os relacionamentos são pai, irmão ou marido, e os nós são tipados pelo sexo da pessoa. Uma representação de um banco de dados orientado a grafos pode ser encontrado na Figura 2.4.

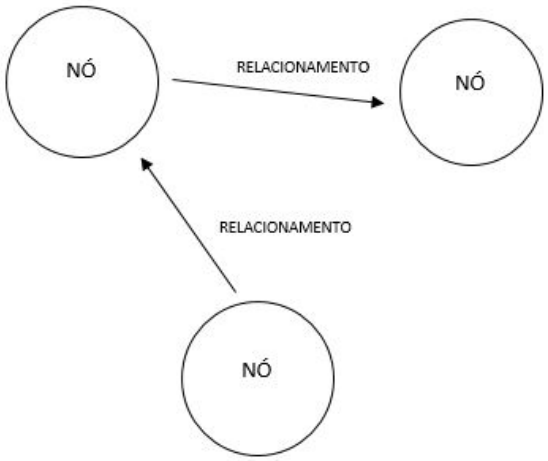

Figura 2.4: Representação da estrutura de um modelo de banco de dados orientado a grafos.

A Teoria de Grafo é um instrumento acessível e poderoso para construção de modelos para inúmeros problemas que requerem a construção de sistemas complexos, que vão desde o mapeamento de processos industriais, logística, sistemas de comunicação, fluxo de redes, escolha de rotas, entre outros. Possuindo ampla aplicação em diversas áreas do conhecimento, tais como, engenharias, computação, genética, física, química, antropologia, linguística.

A ideia intuitiva de um grafo surgiu independente de uma área de conhecimento, no entanto, a Teoria de Grafos é considerada como uma área da matemática aplicada. De acordo com [38], a mais antiga menção sobre o assunto ocorreu no trabalho de Euler, no ano de 1736, para modelar e explicar um problema denominado de Pontes de Königsberg. O problema consistia em verificar se seria possível percorrer todas as sete pontes da cidade passando uma única vez em cada ponte, conforme representada na Figura 2.5. Euler verificou e provou através de um diagrama associando nós (vértices) e arcos (arestas) que não havia solução para o problema, semelhante ao representado na Figura 2.6. É justamente este conceito simples, a essência usada pelo banco de dados orientado a grafos.

Atualmente existem vários tipos de banco de dados orientado a grafos, sendo que cada um desses tipos contam com suas particularidades. Estes também disponibilizam algumas ferramenta ou método para visualização e manipulação dos dados, sendo essas ferramentas bibliotecas implementadas em alguma linguagem de programação ou até mesmo uma linguagem para manipulação e pesquisa. Existem sistemas gerenciadores de bancos de

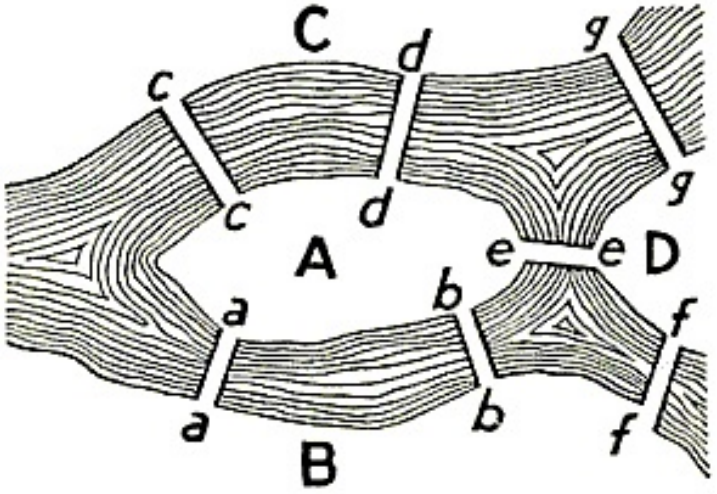

Figura 2.5: Cidade de Königsberg, com suas sete pontes (a-f) [40].

dados orientados a grafos tais como o FlockDB, Neo4j, InfiniteGraph e AllegroGraph (OrientDB). Estes sistemas são muito utilizados atualmente e contam com boa aceitação e suporte da comunidade. Segue a definiçao e algumas particularidades desses sistemas gerenciadores, como abrangência e domínio de utilização e métodos e linguagens.

FlockDB

O FlockDB é um banco de dados criado pelo Twitter com o objetivo de manipular a grande quantidade de informação movimentada diariamente na rede social. De acordo com [31], o FlockDB nasceu com a necessidade de manipulação de dados de uma grande rede social, como criação de contas na rede social, ligações entre os usuários(seguidores, bloqueios), e a grande quantidade de *tweets* postados diariamente.

Este tipo de banco de dados caracteriza-se por ser um banco de dados orientados a grafos simples, que não utiliza operações transversais. Isso quer dizer que ele se diferencia dos outros modelos, no qual é permitido uma pesquisa com mais de um nível. Esta particularidade fica clara na própria rede social Twitter, onde não é necessária a informação dos seguidores de uma pessoa que alguém segue.

O FlockDB armazena os dados em conjuntos de nós e arestas, sendo os nós identificados por inteiros de 64 bits. Em um banco de dados orientado a grafo social, esses *ID's* dos nós representariam a identificação dos usuários, mas no escopo da rede social Twitter esse identificador representa um *tweet*, que é uma espécie de mensagem da rede social Twitter. As arestas também armazenam um número de 64 bits para a triagem do *tweet*, como representado na Figura 2.7.

Por não ser um banco de dados orientado a grafos transversal, o FlockDB leva vantagem quando trata-se de escabilidade e alto número de operações. De acordo com a documentação do FlockDB [39], no Twitter ele é capaz de armazenar 13 bilhões de relacionamentos, executar 20 mil escritas por segundo e realizar 100 mil leituras por segundo. Para manipulação e gerenciamento do banco de dados é disponibilizada uma interface

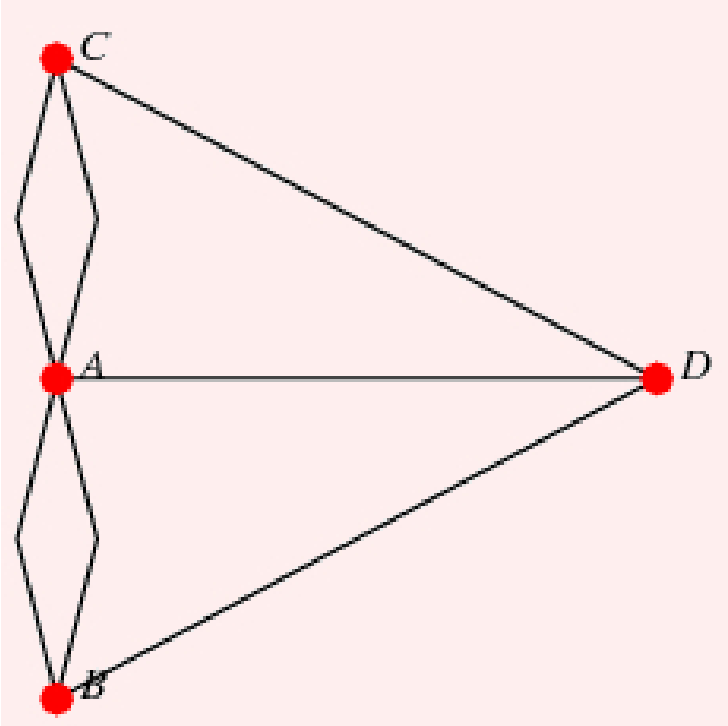

Figura 2.6: Representação das pontes de Königsberg (arestas), interligando os bairros (vértices) [41].

implementada na linguagem de programação PHP, mas esta não é atualizada regularmente, sendo recomendável a utilização de um cliente implementado na linguagem de programação Ruby, como especificado na documentação [39].

InfiniteGraph

O InfiniteGraph é um banco de dados orientado a grafos voltado para modelos empresariais desenvolvido em Java. Este banco de dados caracteriza-se por ajudar os utilizadores a encontrarem relações e informações úteis em uma grande massa de dados. Ele foi implementado e é distribuído pela empresa Objectivity Inc., que o lançou no mercado em 2010. De acordo com a documentação disponibilizada [28], o InfiniteGraph é um banco de dados multi-plataforma, escalável e adequado para lidar com altas taxas de processamento de dados.

De acordo com [16], o InfiniteGraph trabalha muito bem com análise de conexões profundas, como as análises de áreas de inteligência de governos e redes e mídias sociais e empresariais, sendo utilizado em análises financeiras. O InfiniteGraph é capaz de armazenar e processar milhões de nós e arestas de dados. Ele também suporta uma ampla gama de opções para armazenar e acessar os elementos persistentes (arestas e vértices) em seu banco de dados gráfico. Uma aplicação InfiniteGraph pode armazenar dados exclusivamente na máquina local ou distribuir os dados em locais de rede em uma variedade de configurações.

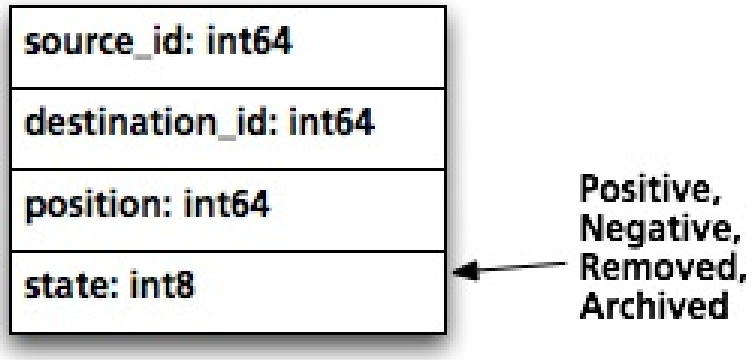

Figura 2.7: Representação da estrutura do FlockDB [29].

O InfiniteGraph também conta com locais de armazenamento flexíveis e configuráveis, além de distribuir a carga de processamento da forma mais eficiente para a aplicação. O usuário pode adicionar locais de armazenamento e zonas para o banco de dados em grafos, a qualquer momento, os tornando imediatamente disponíveis para todos os aplicativos que acessam o banco de dados em grafo. Para atender a grande demanda empresarial, o InfiniteGraph é disponibilizado em várias plataformas, como Linux, Windows e Mac OS e também pode ser implantado em ambientes em nuvem.

AllegroGraph

O AllegroGraph é um banco de dados orientado a grafos de alto desempenho. De acordo com a especificação disponível em [25], o AllegroGraph tem uma utilização eficiente de memória juntamente com armazenamento baseado em disco, permitindo-lhe o escalonamento de bilhões de processos, mantendo um bom desempenho. Desenvolvido nas normas da W3C, o AllegroGraph foi desenvolvido para utilizar como linguagem base e de referência a SPARQL (*Protocol and RDF Query Language*), sendo esta linguagem padronizada para os bancos de dados RDF (*Resource Description Framework*), conhecidos como banco de dados que utilizam triplas.

O AllegroGraph é disponibilizado e implementado pela empresa Franz. Inc, tendo a sua disponibilidade em API's com Java e Phyton. Sua primeira versão foi disponibilizada em 2004. O cliente do AllegroGraph está disponível em Windows, Mac OS X e Linux, sendo multi-plataforma com disponibilidade para arquiteturas 32 e 64 bits. Atualmente, ele é utilizado em projetos *Open Source*, em departamentos de defesa e inteligência, projetos comerciais e em redes sociais como o Twitter, que utiliza-o como ambiente de armazenamento no projeto TwitLogic. O AllegroGraph conta com um navegador próprio, conhecido como Gruff, para a visualização dos dados em grafos e oferece suporte para trabalhar com as linguagens SPARQL e PROLOG.

Neo4j

O Neo4j é um banco de dados de modelo não relacional, orientado a grafos, de código aberto, desenvolvido pela empresa Neo Technology. Alguns artigos citam o banco de dados Neo4j, onde suas funcionalidades e vantagens são apresentadas [23, 32].

Atualmente o Neo4j é o banco de dados orientados a grafos mais popular no mundo. Começou a ser produzido no início dos anos 2000 e foi lançado em 2010, com a versão 1.0, sendo que a versão 2.0 foi lançada em dezembro de 2013. Previamente os bancos de dados em grafos tendem a ter melhor desempenho quando o escopo do banco de dados envolve muitos relacionamentos, pois com uma aplicação do modelo relacional, onde precisaria de uma execução de pesquisa envolvendo vários *joins*, o modelo em grafos poderá executar tal pesquisa sem a necessidade de uma pesquisa complexa, ganhando uma possível vantagem em relação ao modelo relacional quando trata-se de desempenho.

O Neo4j é capaz de armazernar os dados em nós interligados por relacionamentos, assim banco de dados complexos, com bilhões de nós e relacionamentos podem ser manipulados com apenas uma instância do servidor. Mesmo sendo lançado a pouco tempo, o Neo4j vem sendo utilizado há aproximadamente 10 anos. A versão 1.0 veio para fornecer uma estabilidade na API (*Application Programming Interface*), assim melhorando seu desempenho e a capacidade de lidar com milhões de nós e relacionamentos por segundo. O Neo4j também conta com extensões e integrações com linguagens e *frameworks*, tais como Java, Ruby e Phyton, sendo algumas de suas API's encontradas no site desse banco de dados. A estrutura do banco de dados Neo4j é composta por três atributos, conforme apresentado na Figura 2.8:

- nó – corresponde ao vértice do modelo em grafos, cada nó tem um único ID;
- relacionamento – serve para ligar dois nós, assim gerando um relacionamento entre os dois vértices;
- propriedade – é um atributo que pode existir em um nó ou relacionamento, e.g., nome ou idade.

2.3.2 Linguagem

Nesta seção iremos apresentar a linguagem Cypher utilizada em algumas implementações de banco de dados orientado a grafos, e.g., o Neo4j.

Cypher

A linguagem Cypher é utilizada para pesquisa em banco de dados orientado a grafos, sendo muito intuitiva e permitindo uma manipulação eficiente do banco de dados. Construída para ser uma linguagem de alto nível, a linguagem Cypher foi desenvolvida baseada na língua inglesa, assim tornando as *queries* alto-explicativas e intuitivas [27]. O foco principal dessa linguagem é "quê"será pesquisado e não "como"será pesquisado, para chegar a esse objetivo o Cypher foi criado como uma linguagem declarativa.

O Cypher também foi inspirado em uma série de abordagens e baseia-se em práticas estabelecidas para realizar suas consultas. Assim, muitas das palavras e comandos utilizados no Cypher foram inspirados em outras linguagens declarativas de pesquisas em

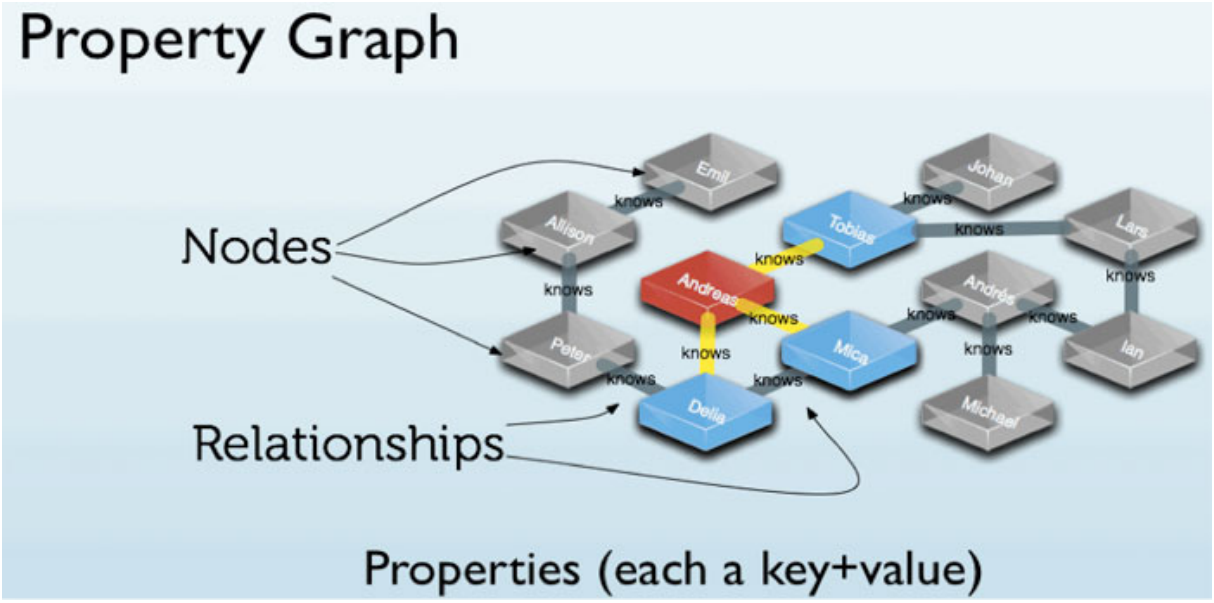

Figura 2.8: Representação da estrutura do Neo4j [24].

banco de dados, como o SQL. Temos como exemplo os comandos *WHERE* e *ORDER BY*. A semântica da linguagem também foi inspirada de outras linguagens, como as linguagens funcionais Haskell e Python. Como citado, o Cypher utiliza algumas estruturas que encontramos também no SQL, tais como:

- MATCH – o padrão da linguagem para fazer correspondência de nós, sendo o meio mais utilizado para recuperar dados do grafo;
- WHERE – não é uma cláusula necessária, mas serve apenas para filtrar um resultado desejado;
- RETURN – estrutura para especificar o que será retornado;
- CREATE – comando para criação de um nó;
- DELETE – comando para a exclusão de um nó;
- SET – define valores para as propriedades dos nós que os nomeiam;
- REMOVE – remove os valores e propriedades dos nós;
- MERGE – atualiza ou cria um novo nó, de acordo com sua existência prévia.

Considere o exemplo de grafo da Figura 2.9. Suponha que queremos executar uma pesquisa que retorne o avô do nó Natan, então executa-se a seguinte consulta:

```
MATCH (neto name: 'Natan')-[:filho]->()-[:filho]->(avo) RETURN neto, avo
```

Tendo como resultado as seguintes informações:

neto	avo
Node[1] name:'Natan'	Node[3] name:'Manoel'

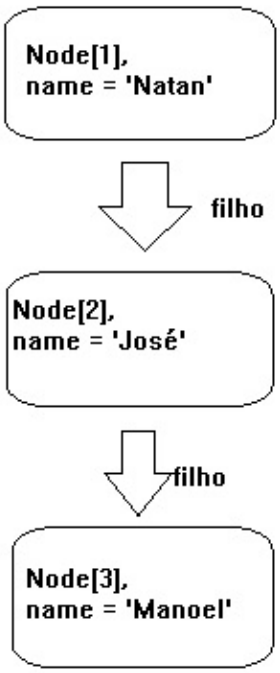

Figura 2.9: Exemplo de um grafo.

2.4 Rede Social Científica

Nesta seção serão discutidos aspectos de rede social científica, iniciando-se com o conceito de rede social com aspectos que envolvem a análise de rede social e a definição de como foi representada a comunidade Brasileira de SI e a rede social do SBSI.

2.4.1 Rede Social

A rede social é uma estrutura de relacionamento composta por um conjunto de pessoas ou atores e um conjunto de relações. Um identificador de uma rede social é o fator em comum entre seus atores, induzindo uma ligação entre eles. Com essa construção básica é possível exemplificar as interações entre vários atores, gerando uma rede social para análise e estudo.

Um fator determinante de uma rede social é o compartilhamento de informações e conhecimento entre os atores com um objetivo em comum entre eles. Exemplificando, uma rede social pode representar um relacionamento de negócios entre empresas ou a amizade entre pessoas de um determinado grupo, entre outras variadas formas de relacionamento social.

De acordo com [13], as redes sociais tem adquirido importância crescente na sociedade moderna. São caracterizadas primariamente pela auto geração de seu desenho, pela sua horizontalidade e sua descentralização. Neste sentido, a intensificação da formação das redes sociais gera um processo de fortalecimento da sociedade, com maior participação democrática e mobilização social.

Assim, a análise de redes sociais, surgiu como uma técnica chave na sociologia moderna. O conceito surgiu na Sociologia e Antropologia Social. No final do Século XX, o termo passou a ser olhado como um novo paradigma das ciências sociais, vindo ser aplicada e desenvolvida no âmbito de disciplinas tão diversas como a antropologia, a biologia, os estudos de comunicação, a economia, a geografia, as ciências da informação, a psicologia social, a sociolinguística e, sobretudo, no serviço social. Em 1954, J. A. Barnes [2] começou a usar o termo sistematicamente para mostrar os padrões dos laços, incorporando os conceitos tradicionalmente usados quer pela sociedade quer pelos cientistas sociais.

Em teoria, na estrutura das redes sociais os atores sociais se caracterizam mais pelas suas relações do que pelos seus atributos (gênero, idade, classe social). Estas relações tem uma densidade variável, a distância que separa dois atores é maior ou menor e alguns atores podem ocupar posições mais centrais que outros. Este fenômeno é explicado por alguns teóricos apontando a existência de laços fortes e fracos e a dos buracos estruturais onde se encontram os atores que não podem comunicar entre si a não ser por intermédio de um terceiro.

Através da análise de uma rede social podemos verificar características e representações da estrutura social analisada. Para executar tal análise é necessária a visualização e representação da rede social, onde é possível a realização de métricas e análises concretas das informações obtidas. Neste trabalho, a representação da rede social será feita por grafos, onde os nós dos grafos representam o atores (pesquisadores da área de SI) e as arestas representam as relações entre esses atores (publicações no SBSI).

2.4.2 Comunidade Brasileira de SI

Para facilitar a modelagem e o desenvolvimento do BD da comunidade Brasileira de SI, foi utilizado neste trabalho os pesquisadores que publicam artigos técnicos no SBSI. A comunidade foi então representada pelos autores, responsáveis pelas publicações de artigos nas edições do SBSI realizados de 2005 a 2014, tendo em vista a impossibilidade de adquirir os dados de 2004.

O SBSI é um evento realizado anualmente, patrocinado pela SBC e sob coordenação da CE-SI/SBC. O evento reúne apresentações de trabalhos científicos e discussão de temas relevantes da área de SI, aproximando pesquisadores, estudante e empresários. Por se tratar de um evento de cunho nacional, as publicações são vinculados a autores de diversas instituições públicas ou privadas. O SBSI nasceu a partir do GT2 da SBC, com o objetivo de fomentar as iniciativas que culminasse na formação da comunidade de SI no Brasil. Na história do evento foram realizadas 11 edições, conforme segue:

- I SBSI, realizado na Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre - RS, 2004;
- II SBSI, realizado Universidade Federal de Santa Catarina (UFSC), Florianópolis - SC, 2005;
- III SBSI, realizado no Centro Universitário Positivo (UnicenP), Curitiba - PR, 2006;
- IV SBSI, realizado no Hotel Novo Mundo (organizado pela UNIRIO), Rio de Janeiro - RJ, 2008;
- V SBSI, realizado na Universidade de Brasília (UnB), Brasília - DF, 2009;

- VI SBSI, realizado na Faculdade Metropolitana, Marabá - PA, 2010;
- VII SBSI, realizado na Universidade Federal da Bahia (UFBA), Salvador - BA, 2011;
- VIII SBSI, realizado na Universidade de São Paulo (USP), São Paulo - SP, 2012;
- IX SBSI, realizado na Universidade Federal da Paraíba (UFPB), João Pessoa - PB, 2013;
- X SBSI, realizado no Hotel Blue Tree Premium (organizado pela UEL), Londrina - PR, 2014;
- XI SBSI, realizado na Universidade Federal de Goiás (UFG), Goiânia - GO, 2015.

2.4.3 Rede Social do SBSI

Para representar a rede social relacionada a comunidade de SI foi utilizado uma rede social científica. Os atributos dessa rede utilizam a mesma representação de uma rede social, onde se utiliza autores e relações entre autores. Na rede social em questão, utilizaremos quatro componentes para construir a rede e representar as publicações: autor, artigo, evento e instituição de vinculação do autor. Esses quatro componentes representam os atores da rede social, assim como o relacionamento entre esses autores, através das relações ligadas ao papel deles na comunidade de SI. Um esboço dessa rede social é apresentado na Figura 2.10, onde um autor produz um artigo, um artigo é publicado em uma edição do SBSI e um autor é vinculado à uma instituição.

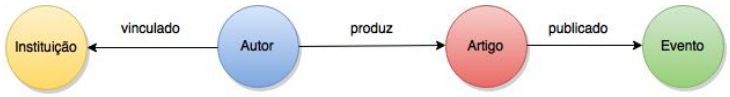

Figura 2.10: Estrutura básica da rede social científica dos autores no SBSI.

2.5 Trabalhos Correlatos

O estudo realizado em [11] e publicado em [12] serviu como base para a realização deste trabalho, pois foi utilizada a base de dados de 2006 a 2011 disponibilizada pelo autor. Em [12], os autores fizeram uma análise de colaboração entre os pesquisadores e instituições participantes do SBSI, utilizando o conceito de corte de vértices em grafos e análise de redes sociais. Os autores utilizaram uma base de dados relacional com as informações dos artigos, autores e instituições, juntamente com suas relações, aplicando algoritmos para possibilitar a análise da rede social com corte de vértices em grafos. Algumas informações relevantes foram identificadas pelo autores. A pesquisa identificou que 25% das instituições não estabeleceram parcerias na publicação de artigos nas edições do SBSI de 2006 a 2011. Outra informação identificada foi que autores com destaque em determinadas edições do SBSI não apresentavam destaque em outras edições do SBSI. Este trabalho auxiliou no processo de conhecimento dos membros da comunidade de SI,

no entanto não foi disponibilizado os dados na Web para facilitar o acesso a informação, diferentemente do objetivo desta monografia.

O trabalho [33] também serviu como base para a realização deste, pois os autores utilizam mineração de dados para realizar uma análise social da rede do SBSI. Os autores utilizaram as publicações do eventos de 2008 a 2011 do SBSI, extraindo as informações dos autores, as instituições de vinculação dos autores e o título de cada artigo. Após a extração das informações foi utilizada a ferramenta *Pajek* a para geração dos grafos de colaboração da comunidade possibilitando a visualização de termos relevantes dos artigos publicados. Os autores também implementaram um sistema de classificação automática alcançando um resultado de aproximadamente 90% dos tópicos de interesse relevante para os autores do SBSI.

Capítulo 3

Proposta de Solução

Para solucionarmos o problema apresentado na Seção 1.1 foi proposta a utilização dos artigos apresentados nas trilhas técnicas das edições de 2005 a 2014 do SBSI para compor um banco de dados relacional e outro orientado a grafos. Desta forma, as informações recuperadas do banco podem representar a rede social científica do SBSI, auxiliando no processo da tomada de decisão da CE-SI e facilitando o aumento de conhecimento dos membros que integram a rede social científica da área de SI no Brasil. Neste capítulo apresentaremos os aspectos metodológicos e de implementação do trabalho realizado.

3.1 Detalhamento Metodológico

Um dos problemas apresentados pela comunidade de SI é a falta de informação disponível em um banco de dados centralizado. Desta forma, foi empregado um método de análise de rede social utilizando um banco de dados conforme proposto em [37], onde utiliza-se um banco de dados em grafo para análise de rede social. Diferente de [37], neste trabalho não foi utilizado unicamente um banco de dados em grafo, pois quando se trata da busca pela melhor alternativa para análise de uma rede social devemos obter várias opções de tratamento dos dados.

De acordo com os métodos empregados por [3], foram escolhidos métodos e modelos de análise, que permitiram a criação de um banco de dados relacional e a criação de um banco de dados modelado em grafos utilizando as informações da rede social científica do SBSI. De acordo com [9] há métodos para se obter conceitos e padrões em estruturas de banco de dados, assim revelando relações entre as partes dessas estruturas. Ainda segundo [9], uma mineração e análise de dados podem ser executadas em estruturas representadas por grafos, sendo assim possível uma análise social em uma rede social representada por grafos.

Considerando os diversos paradigmas de banco de dados, o primeiro modelo adotado neste trabalho foi o banco de dados relacional, uma vez que este modelo é o mais utilizado dentre os banco de dados, e certamente facilita a criação de um repositório centralizado de dados da comunidade de SI. Este paradigma também viabiliza ferramentas para visualização das informações e manipulação dos dados. O sistema de gerenciamento de banco de dados relacional utilizado neste trabalho é o MySQL. De acordo com [30], o MySQL é um sistema de fácil manipulação e é funcional para aplicações de Internet. A escolha

desse SGBD para a confecção deste trabalho foi por ser um sistema multi-plataforma, ter compatibilidade com várias linguagens e ser um software livre com base na GPL [21].

O segundo modelo de armazenamento de dados utilizado foi o orientado a grafos, pois esse modelo tem a sua utilização conhecida em redes sociais. Após a análise dos tipos de BD orientado a grafos apresentados na Seção 2.3.1, foi definido que a melhor opção a ser utilizada neste trabalho é o Neo4j, pois segundo [20], ele conta com extensões para várias linguagens e é um dos únicos que é *open-source* para fins não-comerciais. Desta forma, o sistema de gerenciamento adotado foi o Neo4j, que utiliza a linguagem para consulta Cypher.

Com os bancos de dados definidos e carregados, partiu-se para a análise sobre as informações da comunidade, com visualizações em gráficos. Ainda falta buscar a melhor alternativa de utilização do banco de dados da rede social em termos de visualização. Mas durante a execução deste trabalho foi feita uma comparação e análise de eficiência dos banco de dados desenvolvidos, utilizando métricas e valores de tempo de resposta das pesquisas.

3.1.1 Descrição dos Passos Empregados

Com a finalidade de alcançar os objetivos propostos na Seção 1.2 foi utilizada uma metodologia teórica e prática, que compreende desde o estudo de diferentes abordagens de banco de dados, ferramentas e recursos adequados, até a definição de um modelo de banco que utilize pelo menos duas abordagens para viabilizar a comparação de desempenho dos recursos utilizados.

Considerando a ordem dos passos empregados no desenvolvimento dos bancos de dados, primeiramente, foram feitas as modelagens dos bancos sendo que as características das modelagens são apresentadas nas Seções 3.1.2 e 3.1.3. Após a modelagem, os bancos foram carregados com as informações dos artigos publicados nas edições do SBSI de 2005 a 2014, com os relacionamentos e modelagem adaptados para cada modelo de banco utilizado.

Depois do desenvolvimento dos bancos de dados foi feita a análise dos dados da rede social para obter as respostas das questões propostas na Seção 1. Em seguida, foram comparados os resultados de consultas aplicadas nesses modelos e assim realizada a experimentação, para se obter as características de cada alternativa utilizada.

Os passos descritos são representados na Figura 3.1 para a construção da abordagem relacional e Figura 3.2 para a abordagem em grafo. Note que os passos do desenvolvimento estão representados nas figuras sequencialmente, mas o desenvolvimento de cada abordagem foi executado em paralelo.

Primeiramente, os modelos de banco de dados foram criados paralelamente, sendo representados pelas Figuras 3.4 e 3.5. Após a criação dos modelos de banco de dados utilizados neste trabalho, iniciou-se a etapa de criação do banco relacional, sendo criado manualmente no SGBD MySQL.

Em seguida, foram feitas as inserções das informações no banco relacional. Os dados inseridos são referentes as edições de 2006 a 2011 e foram importados do estudo realizado por [12]. Os dados importados dessa pesquisa estavam disponíveis para importação apenas na linguagem SQL, tornando-se necessário o início da carga ser no modelo relacional criado. Após a importação desses dados, os dados referentes às edições de 2005, 2012, 2013

Figura 3.1: Ordem de execução dos passos empregados na construção da abordagem relacional.

e 2014, foram inseridos manualmente no BD relacional, com o auxílio de uma interface de entrada de dados em PHP que é descrito na Seção 3.2.1, assim facilitando as inserções e consultas necessárias do BD.

A etapa de criação e inserção dos dados no BD Relacional foi executada somente após essas etapas serem executadas no BD Relacional. Com o término da criação e inserções no BD Relacional, a mesma etapa de criação e inserções manuais deveria ser repetida para o banco de dados orientado a grafos, porém por se tornar um processo repetitivo e desgastante, foi criado um script em Java, que é descrito na Seção 3.2.2, que criou o BD Orientado a Grafos e exportou os dados inseridos no BD relacional para o BD Orientado a Grafos.

A mesma base de dados foi gerada para os dois modelos de BD estudados. Com as bases de dados populadas com as informações, foi feita a análise das informações da rede social científica utilizando juntamente as duas abordagens de BD, e após foi executada a análise de desempenho das abordagens de BD. As etapas das análises são detalhadas no Capítulo 4.

3.1.2 Modelo de Banco de Dados Relacional

O modelo relacional criado para o banco de dados utilizado neste trabalho é representado na Figura 3.3. No modelo representado são identificadas as tabelas principais e duas tabelas de ligações para utilização do BD.

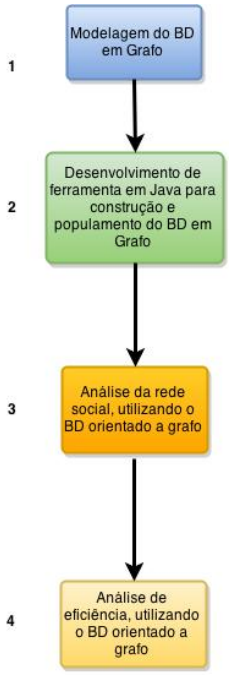

Figura 3.2: Ordem de execução dos passos empregados na construção da abordagem orientada a grafo.

O modelo entidade-relacionamento criado para a confecção do banco de dados relacional é representado na Figura 3.4. No modelo representado, são identificadas quatro entidades principais, Artigo, Autor, Evento e Instituição, e três relacionamentos. As características das entidades que foram recuperadas no estudo de caso, são representadas por atributos.

3.1.3 Modelo de Banco de Dados Orientado a Grafos

Após a criação do banco de dados relacional foi criado o banco orientado a grafos, que recebeu a mesma carga de dados do modelo relacional. Por se tratar de outra abordagem, o modelo foi construído de forma distinta, onde as chaves e identificadores tiveram de sofrer alterações para atender as necessidades desse modelo.

O modelo criado para a base de dados orientada a grafos utilizada neste trabalho é apresentado na Figura 3.5. No modelo orientado a grafos, as entidades que eram representadas em tabelas no modelo relacional, são representadas em vértices e os relacionamentos são representados em arestas, sendo os atributos identificados no modelo relacional representados como propriedades no modelo orientado a grafos, construído de acordo com o padrão de modelo de banco de dados orientado a grafos utilizado pelo Autor [17].

3.2 Aspectos de Implementação

Nesta seção será apresentado o sistema Web para recuperação e armazenamento das informações no banco de dados relacional, bem como um *script* em Java para a migração dos dados do banco de dados relacional para o banco de dados orientado a grafos.

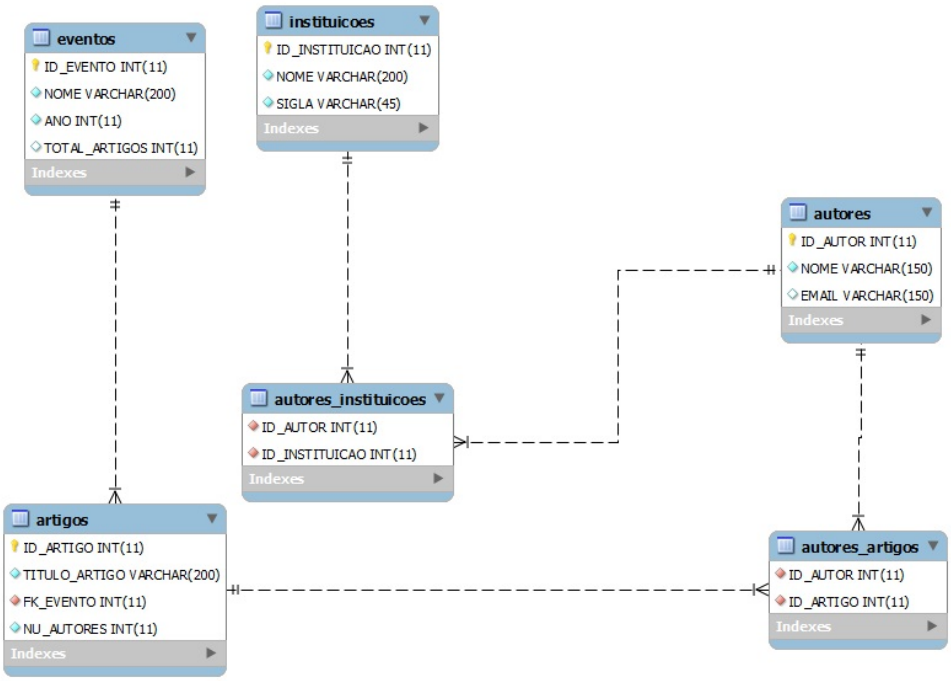

Figura 3.3: Modelo relacional do banco de dados construído.

3.2.1 Sistema Web

Com o intuito de inserir e recuperar as informações no banco de dados relacional, fez-se necessário a criação de um sistema que possibilitasse e auxiliasse essas ações. A inserção dos dados pelo SGBD do MySQL tornaria o processo repetitivo e demorado, pois também seria necessário inserir as ligações das tabelas manualmente. Desta forma, foi disponibilizado na Web uma interface na linguagem SQL, a qual executa as ações de inserção e recuperação.

A Figura 3.6 apresenta a tela inicial da interface, onde é possível cadastrar e recuperar os dados no banco de dados relacional. Na Figura 3.7 está apresentado a listagem resultante da pesquisa de artigos de determinado evento. A Figura 3.8, mostra as informações de determinado artigo, onde é possível ver o evento que foi publicado e os autores. Na Figura 3.9 é possível visualizar as informações de determinado autor, como instituição de vinculação e e-mail.

3.2.2 Script em Java

Para seguir a proposta para a resolução do problema inicial onde é criado um banco orientado a grafos, fez-se necessário a criação de um script para tal ação e tornar o processo de criação e inserção no banco orientado a grafos um processo rápido e que não fosse repetitivo.

O diagrama de classe do script criado é apresentado na Figura 3.10. Primeiramente a classe *ClasseDAO*, faz uma pesquisa de todas as tabelas que existem no banco relacional. Para realizar essa pesquisa é necessária uma conexão com o banco relacional, assim sendo

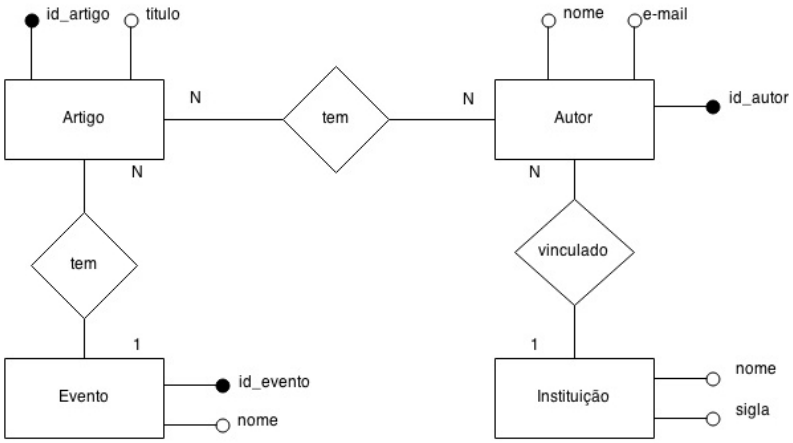

Figura 3.4: Modelo entidade-relacionamento.

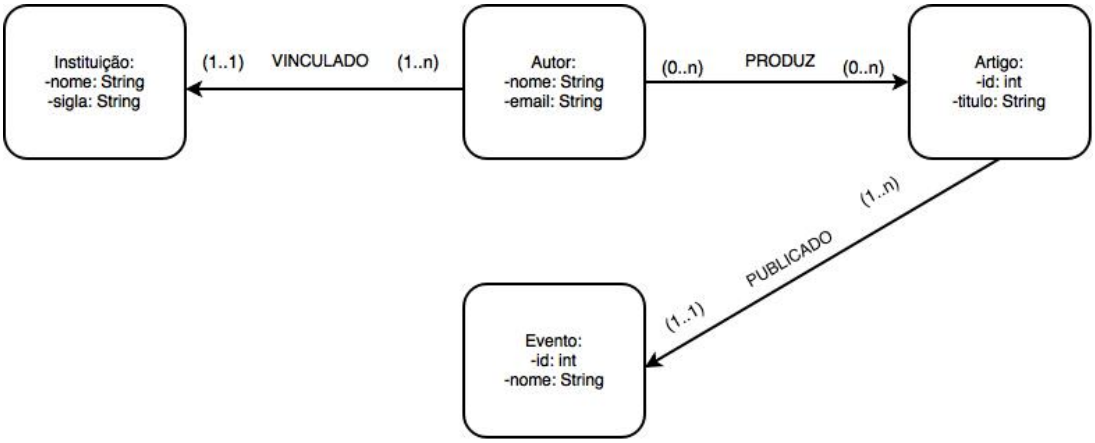

Figura 3.5: Modelo orientado a grafos.

necessário uma chamada da classe *ConexaoMySQL*. A classe de conexão, se conecta ao banco relacional por meio de conectores MySQL que são disponibilizados em [10].

Nesta etapa é realizada a pesquisa, após realizar a conexão com o banco relacional e recuperar as informações da tabela, é realizada a criação desta tabela pesquisada no banco orientado a grafos, e para isso é necessária uma chamada da classe *Neo4j*. A classe *Neo4j*, se conecta ao BD orientado a grafos por meio de conectores disponibilizados em [26].

A classe de criação *Neo4j*, após criar a tabela que lhe foi designada, faz uma chamada da classe *BuildJson*, que cria um arquivo JSON com as informações da tabela, para assim possibilitar a utilização deste arquivo por variados programas de visualização.

Figura 3.6: Tela inicial da interface Web.

Figura 3.7: Listagem de artigos por evento.

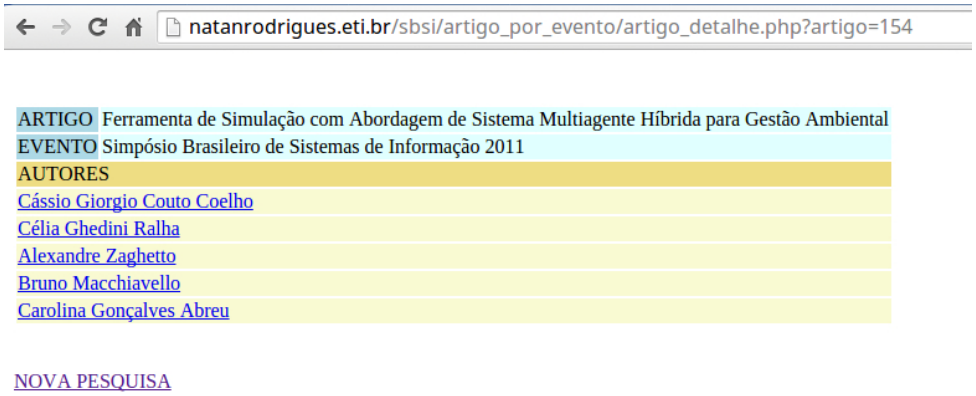

Figura 3.8: Informações de determinado artigo.

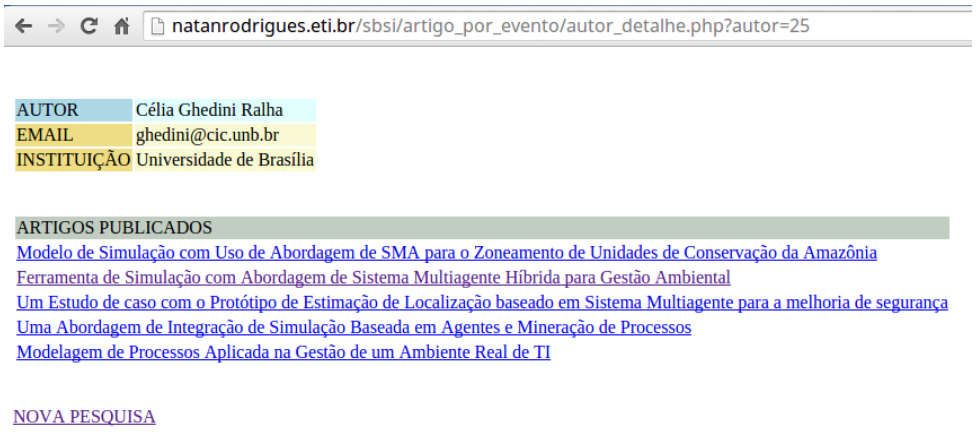

Figura 3.9: Informações de determinado autor.

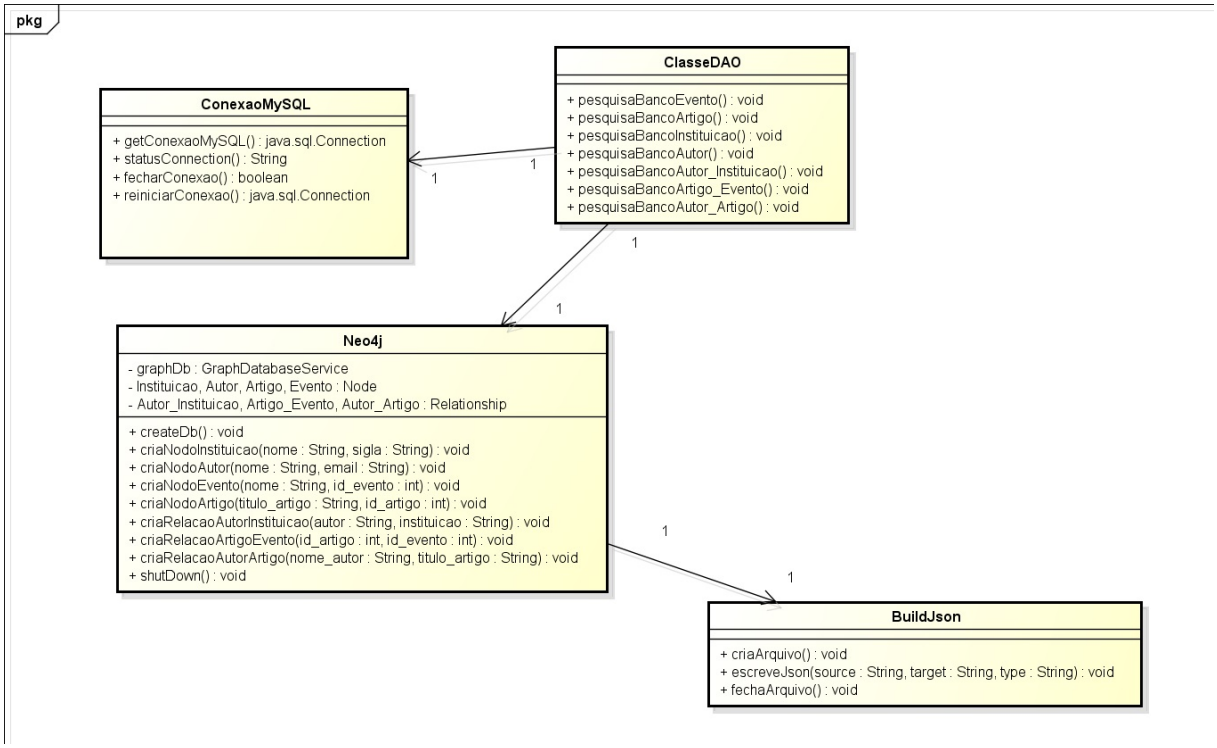

Figura 3.10: Diagrama de classe do Script em Java

Capítulo 4

Experimentação e Análise dos Resultados

Neste capítulo são apresentadas as respostas as questões levantadas no Capítulo 1. Também são analisados os resultados alcançados com as abordagens de banco de dados relacional e orientado a grafos, bem como ilustradas as consultas nas duas abordagens. Finalmente, será apresentada uma análise comparativa de desempenho dos bancos de dados nas duas abordagens.

4.1 Análise das Informações da Comunidade de SI

Com as informações das edições do SBSI 2005 a 2014, armazenadas nos bancos de dados relacional e orientado a grafos, foram feitas algumas consultas e análises que possibilitaram verificar características da rede social científica. Algumas das informações podem ser visualizadas na Tabela 4.1 e nas Figuras 4.1 a 4.3, como a quantidade de artigos publicados, autores e as instituições de vinculação com maior frequência de publicação nas edições do SBSI. Note que na Tabela 4.1 é possível identificar o total de artigos publicados e a quantidade de autores que publicaram em cada edição do SBSI. Observe também, que o total de autores não significa o quantitativo de autores que já publicaram no SBSI, uma vez que um autor pode publicar em várias edições.

Tabela 4.1: Total de artigos e autores por edição do SBSI.

EDIÇÃO	ARTIGOS	AUTORES
2005	33	101
2006	39	109
2008	39	108
2009	29	96
2010	40	114
2011	32	88
2012	76	220
2013	85	240
2014	63	187
TOTAL	436	1263

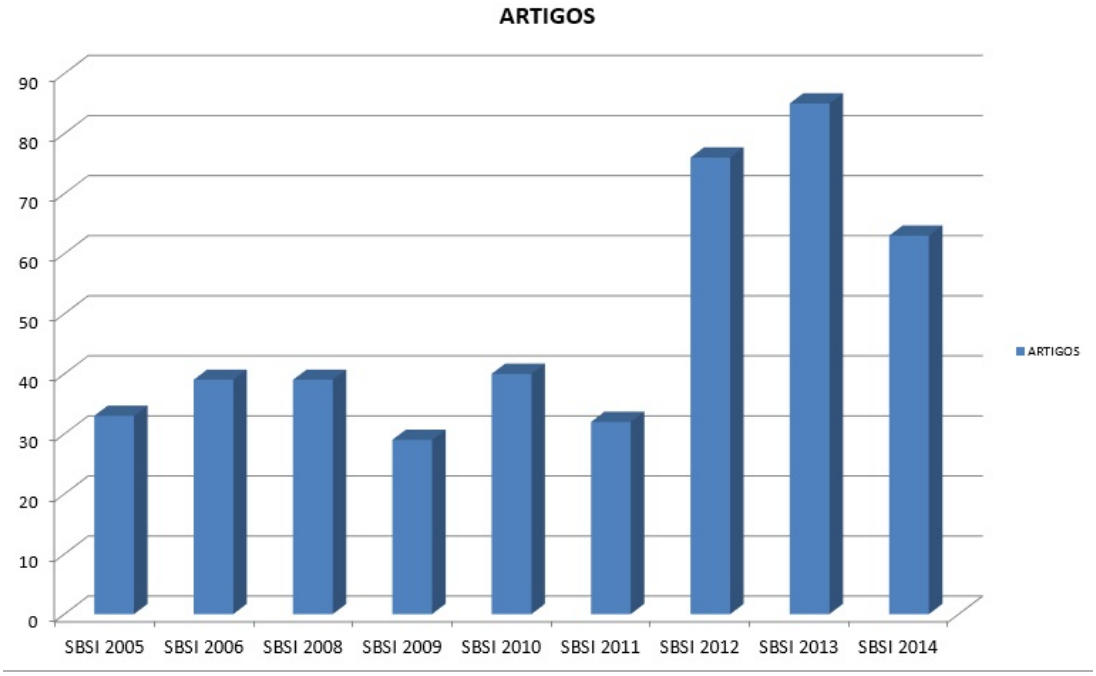

Figura 4.1: Quantidade de artigos publicados nas trilhas técnicas por edição do SBSI.

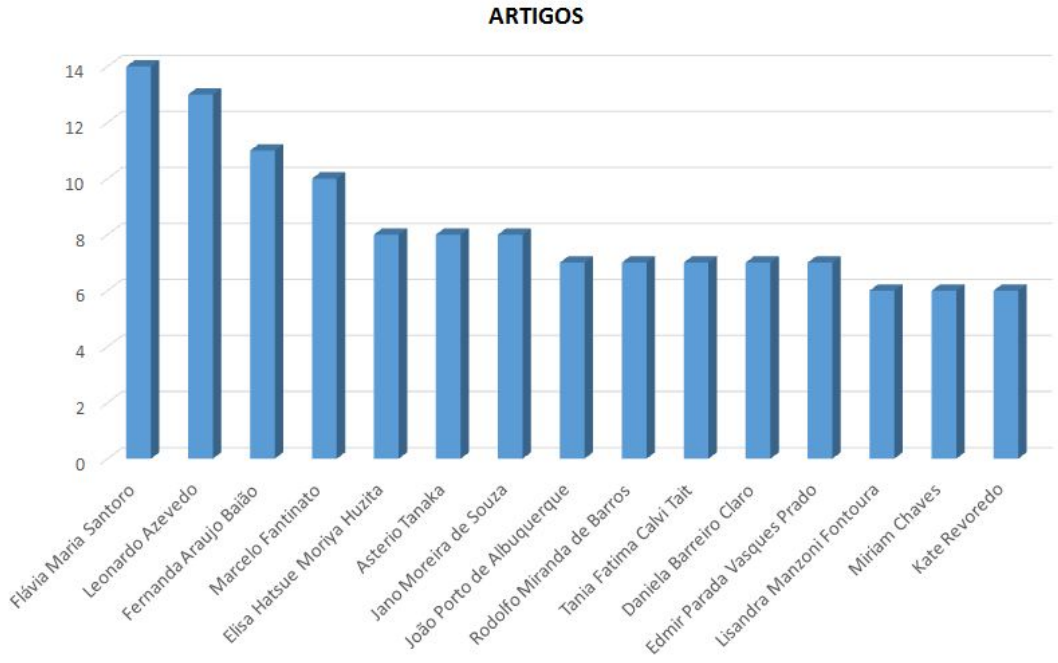

Figura 4.2: Os 15 autores com maior frequência de publicação nas edições do SBSI.

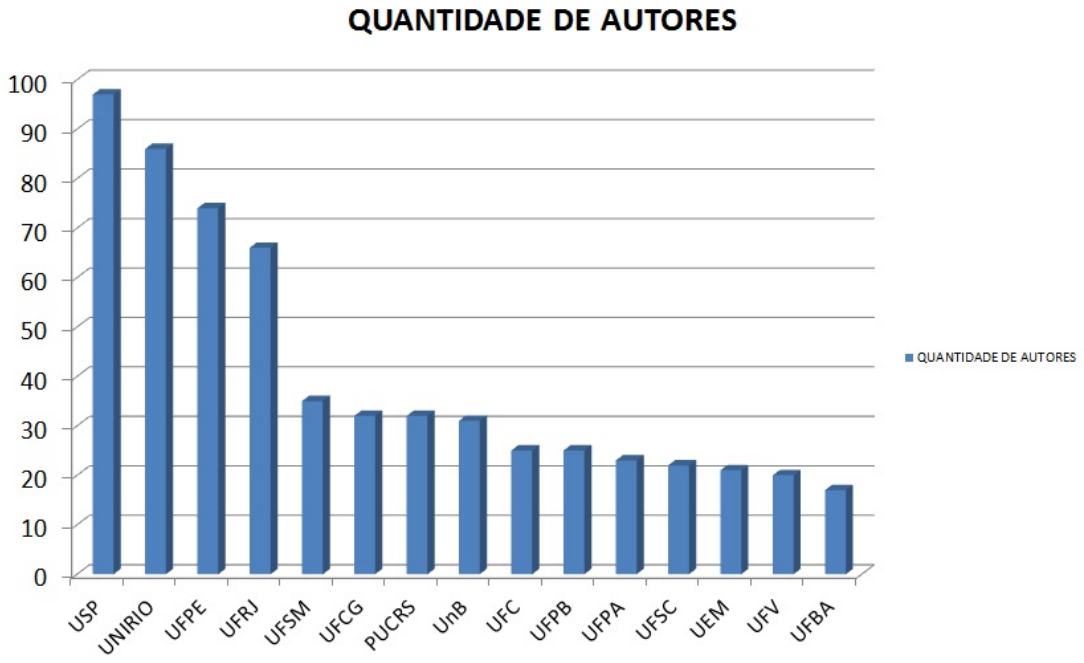

Figura 4.3: As 15 instituições de vinculação dos autores com maior frequência de publicação nas edições SBSI.

Com a análise das informações da comunidade de SI foi possível identificar algumas características da rede social e responder as questões definidas no Capítulo 1, a saber:

- Qual o volume de artigos aceitos nas diversas edições do SBSI (vide Figura 4.1, Listing 4.1 e Listing 4.2);
- Quais os pesquisadores que mais publicam no SBSI (vide Figura 4.2, Listing 4.3 e Listing 4.4);
- Quais as instituições com maior representatividade no SBSI (vide Figura 4.3, Listing 4.5 e Listing 4.6); e
- Com quem estes pesquisadores publicam (vide Figura 4.4, Listing 4.7 e Listing 4.8).

Listing 4.1: Consulta em SQL que Retorna a Quantidade de Artigos Publicados em Cada Edição

```
SELECT count(*), e.NOME
FROM artigos a, eventos e
where a.FK_EVENTO = e.ID_EVENTO
group by e.NOME;
```

Listing 4.2: Consulta em CYPHER que Retorna a Quantidade de Artigos Publicados em Cada Edição

```
MATCH (a:Artigo)-[:publicado]->(e:Evento)
RETURN e.ID, count(a.ID) ORDER BY e.ID;
```

Listing 4.3: Consulta em SQL que Retorna quais os 15 pesquisadores que mais publicam no SBSI

```
SELECT t.nome, count(a.ID_ARTIGO) as QTD_ARTIGOS
FROM artigos a, autores t,autores_artigos ar
where a.ID_ARTIGO = ar.ID_ARTIGO
and t.ID_AUTOR = ar.ID_AUTOR
group by 1
order by 2 desc LIMIT 15;
```

Listing 4.4: Consulta em CYPHER que Retorna quais os 15 pesquisadores que mais publicam no SBSI

```
MATCH (a:Autor)-[:produziu]->(b:Artigo) RETURN a.Nome,count(b.ID)
as QTD_ARTIGO ORDER BY QTD_ARTIGO DESC LIMIT 15;
```

Listing 4.5: Consulta em SQL que Retorna quais as 15 instituições com mais representatividade no SBSI

```
SELECT i.Sigla, i.NOME as INSTITUICAO , count(ai.ID_AUTOR)
as QTD_AUTORES
FROM autores t, instituicoes i, autores_instituicoes ai
where ai.ID_AUTOR = t.ID_AUTOR
and ai.ID_INSTITUICAO = i.ID_INSTITUICAO
group by 2
order by 3 desc LIMIT 15;
```

Listing 4.6: Consulta em CYPHER que Retorna quais as 15 instituições com mais representatividade no SBSI

```
MATCH (a:Autor)-[:vinculado]->(i:Instituicao)
RETURN i.Nome, i.Sigla, count(DISTINCT a.Nome)
as QTD_AUTORES ORDER BY QTD_AUTORES DESC LIMIT 15;
```

Conforme apresentado foram identificados qual o volume de artigos aceitos nas edições do SBSI, quais os pesquisadores que mais publicaram nas edições do SBSI, quais instituições com maior representatividade no SBSI e também com quais pesquisadores um específico autor publica.

Com esta análise também é possível verificar quais os autores já publicaram em co-autoria com um determinado autor. Na Figura 4.4 é possível ver uma representação em grafo dos autores que já publicaram em co-autoria com determinado autor, sendo possível a visualização dos artigos resultantes dessas co-autorias. O Listing 4.7 apresenta a consulta em SQL e o Listing 4.8 em CYPHER.

Também é possível consultar informações específicas. A Figura 4.5 mostra o grafo gerado com as informações de publicação para um determinado autor. A consulta retornou quais os artigos que um determinado autor publicou e em quais edições do SBSI foram publicados. O Listing 4.9 apresenta a consulta em SQL e o Listing 4.10 em CYPHER.

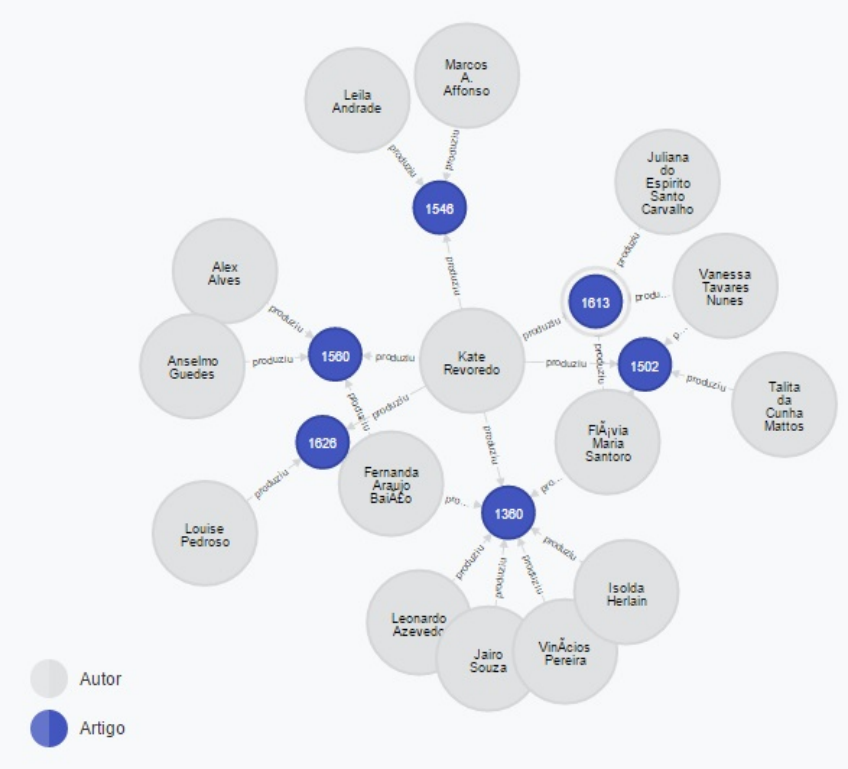

Figura 4.4: Representação em grafo de co-autores de determinado autor.

Listing 4.7: Consulta em SQL que retorna todos os autores que publicaram em co-autoria com determinado autor.

```
SELECT distinct t.nome
FROM artigos a, autores t, autores artigos ar
WHERE a.id artigo = ar.id artigo
and t.id autor = ar.id autor
and a.id artigo in
        (SELECT a.id artigo from
        artigos a, autores t, autores artigos ar
        WHERE a.id artigo = ar.id artigo and
        t.id autor = ar.id autor
        and t.nome = 'Fulano de Cicrano');
```

Listing 4.8: Consulta em CYPHER que retorna todos os autores que publicaram em co-autoria com determinado autor.

```
MATCH (a:Autor)-[:produziu]->(b:Artigo)
<-[:produziu]-(d:Autor)
WHERE a.Nome = 'Fulano de Cicrano'
RETURN DISTINCT d.Nome;
```

Figura 4.5: Representação em grafo das publicacões de um autor específico.

Listing 4.9: Consulta em SQL que retorna todos os artigos de um autor e quais edições do SBSI foram publicados.

```
SELECT t.NOME ,a.TITULO_ARTIGO, e.NOME
FROM artigos a, autores t, autores_artigos ar, eventos e
WHERE a.id_artigo = ar.id_artigo
AND t.id_autor = ar.id_autor
AND t.NOME = 'Fulano_de_Cicrano'
AND a.fk_evento = e.id_evento;
```

Listing 4.10: Consulta em CYPHER que retorna todos os artigos de um autor e quais edições do SBSI foram publicados.

```
MATCH (a:Autor)-[:produziu]->(b:Artigo)<-[:produziu]-(d:Autor)
WHERE a.Nome = 'Fulano de Cicrano' RETURN DISTINCT d.Nome;
```

O conceito de profundidade que é utilizado para descrever a consulta é o mesmo conceito de profundidade de árvores definido por [36]. A consulta de co-autoria é uma consulta em profundidade, pois a consulta verifica o nó ou a entidade autor mais de uma vez. De acordo com [36], toda árvore pode ser representada por um grafo. Como a consulta de co-autoria é iniciada em um nó de autor que busca outro nó autor, podemos representá-la em uma estrutura de dados de árvore.

A consulta apresentada na Figura 4.4, é uma consulta de profundidade 1, pois a partir de um Autor X o retorno da consulta é outro Autor Y. Se a consulta retornar um Autor Z a partir de um Autor Y resultante de outro Autor X, a consulta teria profundidade 2. Assim, pode-se definir que $P = Q - 1$, onde P é a profundidade da consulta, e Q é quantidade de vezes que a consulta percorre a entidade autor.

4.2 Análise de Desempenho dos Bancos de Dados

Com a análise dos dados da comunidade de SI que publicaram nas diversas edições do SBSI foi possível realizar a análise dos bancos de dados e verificar qual abordagem de

banco nos oferece melhor desempenho.

Para esta análise foram realizadas dez vezes a mesma consulta no banco, registrando o tempo de cada execução, e as métricas foram realizadas com "cache frio"e sem a utilização de índices. A ferramenta *MySQL Workbench 6.2* foi utilizada para realizar as consultas e recuperar os tempos de execução no banco relacional. Para execução das consultas e recuperação dos tempos de execução do banco orientado a grafos, foi utilizada a ferramenta *Neo4j Community Edition*. A média dessas métricas foi considerada como o tempo de cada consulta.

Para análise de eficiência dos bancos foram executadas três consultas de profundidade zero, uma consulta de profundidade 1, uma consulta de profundidade 2, uma consulta de profundidade 3, uma consulta de profundidade 4 e uma consulta de profundidade 5.

Na análise comparativa da eficiência de duas abordagens de banco, foi possível verificar que para as consultas de profundidade 0 e 1, o banco relacional tem uma pequena vantagem de eficiência. Quando a consulta se estende para profundidade 2 os dois tipos de banco têm basicamente o mesmo tempo de eficiência. A partir da profundidade 3 o banco orientado a grafos tem um menor tempo de consulta, com uma concreta vantagem no tempo de consulta a partir da profundidade 4, podendo ser visualizado na Figura 4.6.

A Tabela 4.2 mostra os tempos médios das diversas profundidades de consulta considerando as duas abordagens de banco - relacional e grafos.

Tabela 4.2: Tempo em milissegundos da execução das consultas em cada abordagem de banco de dados.

Profundidade	Relacional (ms)	Grafos (ms)
0	0,007	0,024
0	0,028	0,046
0	0,051	0,068
1	0,028	0,062
2	0,078	0,080
3	1,568	0,979
4	23,762	10,207
5	45,391	10,435

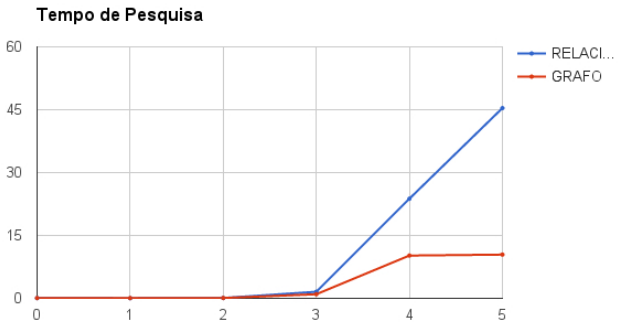

Figura 4.6: Tempo de consulta em profundidade das duas abordagens de banco de dados.

Capítulo 5

Conclusões e Trabalhos Futuros

Este trabalho apresentou uma análise da comunidade de SI utilizando as informações geradas pelos artigos publicados nas trilhas técnicas principais das edições de 2005 a 2015 do SBSI. Com a manipulação dessas informações, foi realizada uma análise comparativa entre duas abordagens de banco de dados, sendo uma abordagem relacional e a outra orientada a grafos. Os objetivos definidos no Capítulo 1 e Seção 1.2 foram alcançados, pois as questões levantadas no Capítulo 1 foram respondidas com as análises realizadas.

Sobre a análise de desempenho, foi realizado um estudo para se definir a melhor abordagem de banco de dados a se utilizar em uma rede social científica, onde a realização da análise de eficiência nos ajudou a levantar características e verificar informações do funcionamento dessas abordagens distintas. Para o banco orientado a grafos, foi verificado que para redes onde é necessário realizar consultas a partir de 4 níveis de profundidade, que necessite do cruzamento de dados de uma mesma entidade ou tipo de nó, como autor, é uma melhor alternativa em comparação com o modelo relacional, pois foi verificado que o tempo de consulta é menor. Pois no banco relacional esta consulta necessita de várias junções entre as entidades, diferente do banco orientado a grafos que utilizam os relacionamentos entre os grafos para realizar este tipo de consulta.

O banco de dados desenvolvido neste trabalho está disponível para a comunidade no sítio `http://www.natanrodrigues.eti.br/sbsi`, onde várias consultas poderão ser executadas, servindo para recuperar informações que podem ser necessárias para auxiliar a tomada de decisão da CE-SI e aumentar o conhecimento dos membros da comunidade Brasileira de SI, servindo para outros estudos sobre a comunidade Brasileira de SI. Também foi publicado no SBSI 2015 um artigo para comunicação dos resultados desta pesquisa, o qual foi intitulado: Conhecendo a Comunidade de Sistemas de Informação no Brasil: um Estudo Comparativo Utilizando Diferentes Abordagens de Banco de Dados. [14]

Como proposta de trabalhos futuros pode-se citar a aplicação deste estudo em redes sociais de maior escala, juntamente com a disponibilização de uma ferramenta para a pesquisa das informações levantadas na rede social científica do SBSI. Também seria necessário que quando as consultas fossem executadas, uma visualização das estruturas em grafo fossem geradas, facilitando o entendimento do escopo da consulta e a análise dos resultados. Outra sugestão é automatizar a inserção de informações das futuras edições do SBSI para manter o banco de dados atualizado com a realização de cada evento do SBSI.

Referências

[1] Sociedade brasileira de computação - sbc / sistemas de informação. Disponível em: http://www.sbc.org.br/index.php?option=com_content&view=category&layout=blog&id=298&Itemid=917. Acessado em 02 de Junho de 2015. 2

[2] J. A. Barnes. Class and Committees in a Norwegian Island Parish. *Human Relations*, 7(1):39–58, February 1954. 22

[3] S. Bordoloi and B. Kalita. Article: Designing graph database models from existing relational databases. *International Journal of Computer Applications*, 74(1):25–31, July 2013. 14, 25

[4] D. Chamberlin and R. Boyce. Sequel: A structured english query language. *Proc. ACM SIGMOD Workshop on Data Description*, 1974. 11

[5] P. Chen. The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, March 1976. 4

[6] F. E. Codd. Relational model of data for large shared data banks. *IBM Research Report, San Jose, California*, 13(6). 7

[7] F. E. Codd. Does your dbms run by the rules. *ComputerWorld*, October 21, 1985. 7

[8] F. E. Codd. Is your dbms really relational? *ComputerWorld*, October 14, 1985. 7

[9] D. J. Cook and L. B. Holder. Graph-based data mining. *IEEE Intelligent Systems*, 15:32–41, 2000. 25

[10] Oracle Corporation. Mysql :: Download connector/j. Disponível em: http://dev.mysql.com/downloads/connector/j/. Acessado em 02 de Abril de 2015. 30

[11] E. A. de Oliveira. Sobre a colaboração na comunidade de sistemas de informação através dos simpósios SBSI. Dissertação, Centro de Ciências Exatas e Tecnologia, UNIRIO, 2012. 23

[12] E. A. de Oliveira and V. M. F. Dias. Redes sociais do sbsi e o corte de vértices como base para identificar atores importantes na coesão de grupos de pesquisa. *Simpósio Brasileiro de Sistemas de Informação 2012 (SBSI)*, 2012. 23, 26

[13] Centro de Processamento de Dados do Estado de Mato Grosso. Rede social: característica, estrutura e importância social. Disponível em: http://www.cepromat.mt.gov.br/apresenta-noticias/-/asset_publisher/MP02bJ29G7PZ/content/

rede-social-caracteristica-estrutura-e-importancia-social, Acessado em 07 de Julho de 2015. 21

[14] N. de S. Rodrigues and C. G. Ralha. Conhecendo a comunidade de sistemas de informação no brasil: um estudo comparativo utilizando diferentes abordagens de banco de dados. In *Anais XI Simpósio Brasileiro de Sistemas de Informação (SBSI)*, pages 576–582, Goiânia, GO, 2015. Association for Computing Machinery (ACM). 41

[15] S. Edlich. Nosql databases. Disponível em: http://nosql-database.org/, Acessado em 13 de Julho de 2015. 14

[16] R. Empson. Infinitegraph steps out of beta to help companies identify deep relationships in large data sets. August 2014. Disponível em: http://techcrunch.com/2011/08/16/infinitegraph-steps-out-of-beta-to-help-companies/identify-deep-relationships-in-large-data-sets. Acessado em 07 de Junho de 2015. 17

[17] G. C. G. Van Erven. Mdg-nosql: Modelo de dados para bancos nosql baseados em grafos. Dissertação, Instituto de Ciências Exatas, Departamento de Ciência da Computação, UnB, 2015. 28

[18] f. Machado and M. Abreu. *Projeto de Banco de dados: Uma visão prática*. Érica. 4, 6, 8, 9

[19] J. Filipe, J. Soares, and T. Coutinho. Sistema de gerenciamento de restaurantes. Disponível em: http://www.cin.ufpe.br/~tcs5/aps/files/modelo_relacional.jpg. Acessado em 08 de Julho de 2015. x, 10

[20] K. Finley. 5 graph databases to consider, Abril de 2011. Disponível em: http://readwrite.com/2011/04/20/5-graph-databases-to-consider, Acessado em 3 de Fevereiro de 2015. 26

[21] Free Software Foundation. The gnu general public license v3.0 - gnu project. Disponível em: http://www.gnu.org/copyleft/gpl.html. Acessado em 3 de Fevereiro de 2015. 26

[22] C. A. Heuser. *Projeto de Banco de dados: Uma visão prática*. Sagra Luzzato, 1998. 5, 6

[23] M. Hunger. Neo4j - base de dados nosql baseada em java. March 2010. Disponível em: {http://www.infoq.com/br/news/2010/03/neo4j-10}. Acessadoem05deJunhode2015. 19

[24] M. Hunger. Why choose a graph database. May 2015. Disponível em: http://radar.oreilly.com/2013/07/why-choose-a-graph-database.html. Acessado em 08 de Junho de 2015. x, 20

[25] Franz Inc. Allegrograph. October 2014. Disponível em: http://franz.com/agraph/allegrograph. Acessado em 07 de Junho de 2015. 18

[26] Neo Technology Inc. 35.1. include neo4j in your project - - the neo4j manual v2.2.2. Disponível em: http://neo4j.com/docs/stable/tutorials-java-embedded-setup.html. Acessado em 03 de Abril de 2015. 30

[27] Neo Technology Inc. Cypher query language. September 2014. Disponível em: {http://docs.neo4j.org/chunked/milestone/cypher-query-lang.html}. Acessadoem06deJunhode2015. 19

[28] Objectivity Inc. Infinitegraph. October 2014. Disponível em: http://www.objectivity.com/infinitegraph. Acessado em 07 de Junho de 2015. 17

[29] Twitter Inc. Introducing flockdb. Disponível em: https://g.twimg.com/imported-images/bc01508281.png. Acessado em 06 de Junho de 2015. x, 18

[30] J. A. N. G. Manzano. *MySQL 5.1 Interativo - Guia Básico de Orientação e Desenvolvimento*. Érica. 10, 13, 25

[31] M. Marr. Exploration of nosql: Flockdb. March 2014. Disponível em: http://www.devwebpro.com/exploration-of-nosql-flockdb. Acessado em 08 de Junho de 2015. 16

[32] I. Robinson, j. Webber, and E. Eifrem. *Graph Databases*. O'Reilly Media, Inc., 2013. 14, 19

[33] B. A. Silveira and T. Y. Muramatsu. Análise do perfil de uma comunidade científica através de mineração de texto. Monografia, Centro de Ciências Exatas e Tecnologia, UNIRIO, 2011. 24

[34] Strozzi.it. Nosql: a non-sql rdbms. September 2014. Disponível em: http://www.strozzi.it/cgi-bin/CSA/tw7/I/enUS/nosql. Acessado em 06 de Junho de 2015. 13

[35] S. Tahaghoghi and H. E. Williams. *Learning MySQL*. Learning Series. O'Reilly Media, Inc., 2007. 13

[36] A. A. Tenenbaum, Y. Langsam, and Moshe J. Augenstein. *Data Structures Using C*. McGraw-Hill Inc., 1989. 39

[37] Ł. Warchał. Using neo4j graph database in social network analysis. *Studia Informatica*, 33(2A):271–279, 2012. 25

[38] E. W. Weisstein. "königsberg bridge problem"from mathworld–a wolfram web resource. Disponível em: http://mathworld.wolfram.com/KoenigsbergBridgeProblem.html. Acessado em 07 de Julho de 2015. 15

[39] R. Wight. Introducing flockdb. September 2014. Disponível em: https://blog.twitter.com/2010/introducing-flockdb. Acessado em 08 de Junho de 2015. 16, 17

[40] Inc. Wolfram Research. Königsberg bridge problem. Disponível em: http://mathworld.wolfram.com/images/gifs/koenigsb.gif. Acessado em 02 de Junho de 2015. x, 16

[41] Inc. Wolfram Research. Königsberg bridge problem. Disponível em: http://mathworld.wolfram.com/images/eps-gif/KoenigsbergBridges_901.gif. Acessado em 02 de Junho de 2015. x, 17